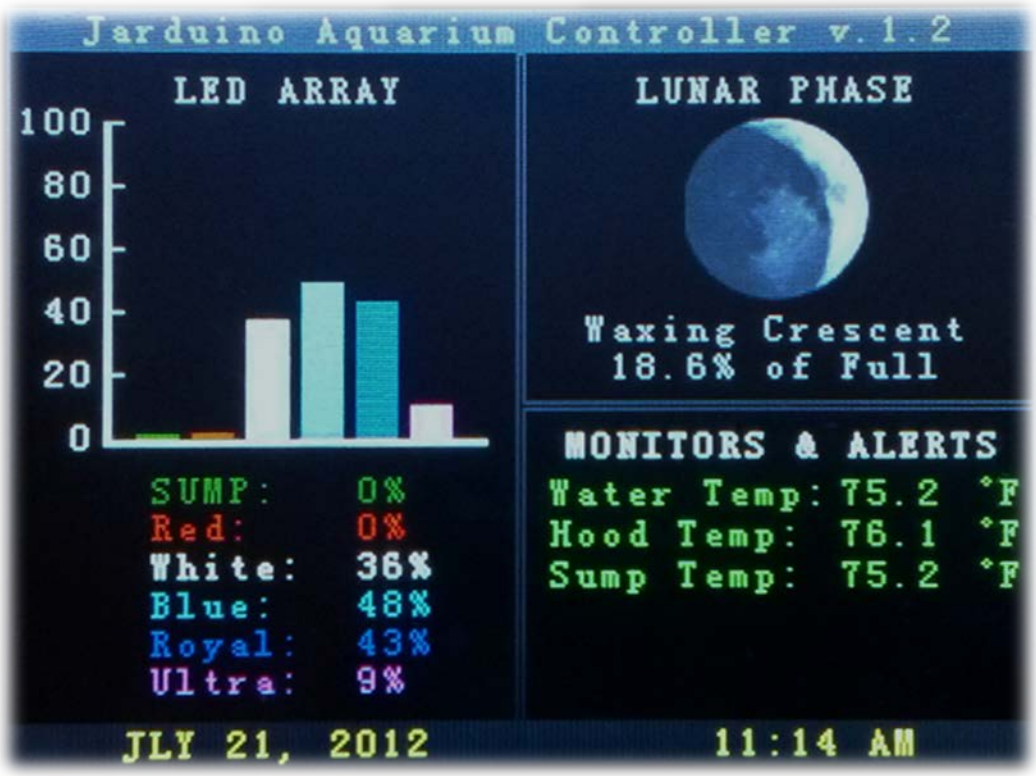
JARDUINO AQUARIUM CONTROLLER

MANUAL de USUARIO

Versión 1.2 beta



BY JAMIE JARDIN

Traducido por Isac Paul Antonio

LICENCIA

Este documento está bajo licencia Creative Commons Attribution-NonCommercial-NoDerivs 3.0 Unported (CC BY-NC-ND 3.0).

PROPÓSITO DEL DOCUMENTO

En este manual de usuario se presenta una descripción del hardware necesario para construir la parte principal del Jarduino y como ensamblarlo. Además está incluido un tutorial paso a paso para cargar el programa en un Arduino MEGA 2560 y una descripción detallada de la funcionalidad del Jarduino Aquarium Controller

HISTORIAL DE CAMBIOS

VERSION	DATA	AUTOR	SUMARIO DE LOS CAMBIOS
1.2 beta	Septiembre 2012	Jamie Jardin	Initial Draft

NORMA LEGAL

Al igual que con cualquier proyecto que utilice los elementos que cortan, queman, caen, giran, azotan, provocan ceguera, etc, la cautela es lo más importante. Por favor, tenga en cuenta que yo no asumo absolutamente ninguna responsabilidad de sus acciones con respecto al uso de cualquier material descrito en este manual.

Los artículos y la información proporcionados son para propósitos educativos solamente. No hay sustitución para las instrucciones del fabricante oficial y el asesoramiento profesional. Por favor, póngase en contacto con el fabricante del producto antes de modificar cualquier dispositivo o software. Por favor, póngase en contacto con un profesional autorizado antes de intentar cualquier cosa física o siguiendo los consejos dados aquí. Ni el autor, Jamie Jardin, ni los demás contribuyentes son responsables por los daños incurridos siguiendo las instrucciones o asesoramiento proporcionado.

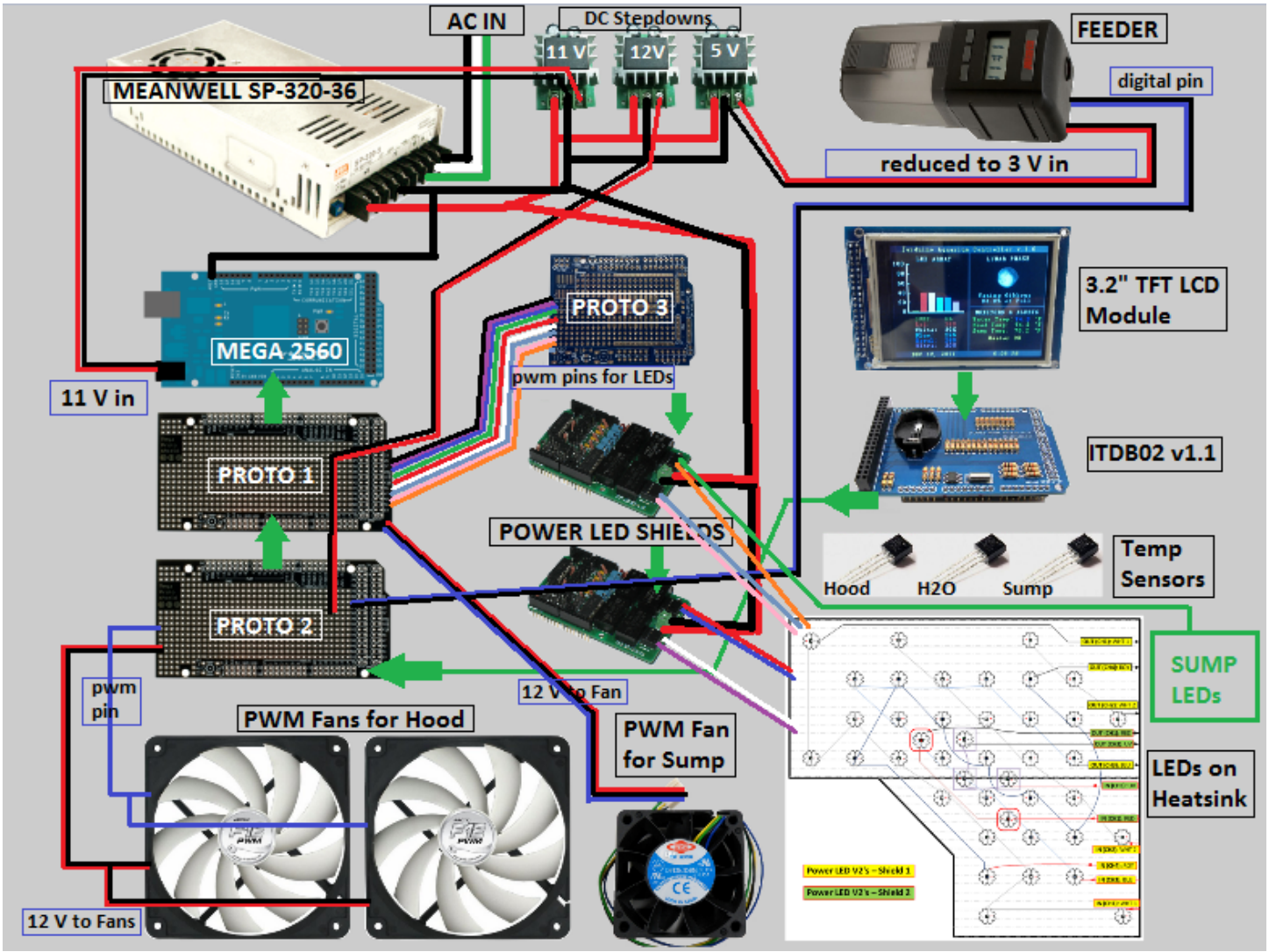
Aunque sería difícil de hacer, si siguiendo estos artículos produces un cortocircuito eléctrico que quema tu casa - no es mi problema. Si la mujer te pide el divorcio poco después, no es mi problema. Si pierdes tu trabajo y terminas en la calle - no es mi problema. Si tu cuenta bancaria está vacía - no es mi problema. Si tus tarjetas de crédito están bloqueadas, yo podría compartir tu dolor, pero todavía no es mi problema. Si sigues adelante con esto, todas las consecuencias son tu propio problema y a mí no se me puede hacer responsable de ninguna manera.

TABLE OF CONTENTS

1. INTRODUCCIÓN ..... - 3 -
CARACTERISTICAS PRINCIPALES (Jarduino Aquarium Controller v.1.1): ..... - 4 -
CARACTERISTICAS PRINCIPALES – ACTUALIZACIONES (Jarduino Aquarium Controller v.1.2 beta) ..... - 4 -
Actualización esperada (Jarduino v.1.3) ..... - 5 -
POSIBLES DESARROLOS ULTERIORES ..... - 5 -
2. INSTRUCCIONES PARA CONSTRUIR EL JARDUINO ..... - 6 -
HARDWARE ..... - 6 -
LISTA DE COMPONENTES ..... - 7 -
Componentes necesarios para el controlador Jarduino: ..... - 7 -
COMPONENTES RECOMENDADOS para el controlador Jarduino: ..... - 10 -
❖ COMPONENTES PARA EL SENSOR DE TEMPERATURA ..... - 10 -
❖ MODULO RELOJ DE TIEMPO REAL ..... - 10 -
❖ COMPONENTES RELOJ DE TIEMPO REAL (RTC) ..... - 11 -
❖ VARIOS COMPONENTES ..... - 12 -
COMPONENTES OPCIONALES para el controlador Jarduino: ..... - 13 -
❖ COMPONENTES PARA ALARMA AUDIO ..... - 13 -
❖ COMPONENTES PARA EL GENERADOR DE OLAS ..... - 14 -
❖ MAS COMPONENTES VARIOS ..... - 15 -
MONTAJE DEL JARDUINO ..... - 16 -
ASIGNACIÓN DE PINES ..... - 17 -
CONECCIÓN DEL RTC ..... - 18 -
CONECCIÓN DE LOS SENSORES DE TEMPERATURA ..... - 19 -
CONECCIÓN DE LOS VENTILADORES ..... - 20 -
CONECCION DE LOS LED ..... - 21 -
3. INSTRUCCIONES DE SOFTWARE ..... - 23 -
SOFTWARE ..... - 23 -
VERSIONES JARDUINO ..... - 23 -
CARGANDO EL PROGRAMA JARDUINO EN EL ARDUINO ..... - 23 -
PASO 1: Instalar el archivador de ficheros ..... - 23 -
PASO 2: Descargar e instalar el software de desarrollo Arduino ..... - 24 -
PASO 3: Conecta tu Arduino MEGA 2560 ..... - 26 -
PASO 4: Configurar el software de desarrollo Arduino ..... - 27 -
PASO 5: Poner todos los archivos en los lugares adecuados ..... - 30 -
PASO 6: BORRAR LA MEMORIA EEPROM ..... - 31 -
PASO 7: Obtener los números de serie de los sensores de temperatura ..... - 33 -
PASO 8: Cargar el código del Jarduino ..... - 35 -
4. JARDUINO FUNCTIONALIDAD ..... - 39 -

1. INTRODUCCIÓN

El Jarduino Aquarium Controller es un controlador basado en Arduino, con una pantalla táctil color de 3.2". Gracias a su interfaz gráfica fácil de utilizar es capable de controlar y monitorizar LEDs, ventiladores, y bombas de movimiento, así como controlar un calentador, un enfriador, una alarma, y un comedero automático. También se podrá ampliar con relativa facilidad. El código está inspirado en los proyectos Krusduino y Stilo . Una diagrama de conexión de las partes electrónicas y los accesorios esta presentada abajo.



NOTA: La diagrama de arriba es una descripción de mi montaje personal. No contiene algunos periféricos que monitoriza y controla el Jarduino, y este manual no cubre cómo conectar todo lo que esta descrito. Para unos detalles más completos sobre mi montaje, (cómo empecé, mis razones para hacer lo que hice, cómo me enganchó todo, y otros detalles), por favor refiérase a mi hilo disponible en UKreefs. Mi "usuario" en los foros es "TheDOdblG".

CARACTERISTICAS PRINCIPALES (Jarduino Aquarium Controller v.1.1):

- Controla hasta 6 canales de LED como Blanco, Azul, Azul Royal, UV, Rojo (o cualquier elección de color)
- Controla un sexto canal para la luz LED del Sumidero/Refugio
- Controla un séptimo canal que simula un ciclo lunar en tiempo real
- Avanzadas funciones de prueba de LED y simulaciones
- Sensores de temperatura para agua y disipador térmico de los LED
- Control de un calentador y un enfriador
- Alarmas audio y visual para variaciones de temperatura predefinidas
- Control dinámico de velocidad para los ventiladores de disipador (luz acuario y refugio)
- Generador de olas, control de bombas de movimiento (varios modos)
- Comedero automático (Programa hasta 4 comidas diarias o da de comer al instante)
- Formato seleccionable por el usuario
    - Escalas de temperature Celsius & Fahrenheit Temperature Scales
    - Formato de hora 24HR & 12HR
    - Formato de fecha MM, DD, YYYY & DD/MM/YYYY Date Formats
- Ajustes y preferencias guardadas en EEPROM (Memoria del Arduino)

CARACTERISTICAS PRINCIPALES – ACTUALIZACIONES (Jarduino Aquarium Controller v.1.2 beta)

- Añadido soporte para Arduino 1.0 IDE (Tienes que utilizar mi versión parcheada)
- Totalmente compatible con Arduino 1.0.1 IDE (No tiene que utilizar mi versión)
- Puede escoger entre Librería ITDB02_Graph16.h o UTFT.h de Henning Karlsen
    - Todos los controladores compatibles (y módulos) están ahora incluidos en el programa
    - Aproximadamente 40 diferentes módulos LCD para elegir!
- Reemplazado la Librería de Matt Joyce DS1307 con la de Henning Karlsen DS1307
    - Modificación de formato en la Librería de Karlsen DS1307
    - Reescritura de todo el código para RTC en el programa, basado en "ITDB02 Analog Clock" de Karlsen
    - Añadido "Día de la Semana" en "Fecha y Hora."
    - Se ha corregido la posibilidad de fijar una fecha que no existe (ex. FEB 31, 2011)
    - Se ha corregido un pequeño bug en la barra “Fecha y Hora”
- Añadido el formato de 24H en la pantalla "Ajustes-Prueba de intensidad LED pantalla"
- Modificado la pantalla "Ver/Cambiar valores LED Luna Min y Max"
    - Añadido fotos y la posibilidad de establecer y guardar la iluminación Min y Max
- Reemplazado el "Widget" de la pantalla “Control individual de LEDs” con "Slider Bars"
- Mejorado la pantalla "Cambiar los valores de LED" con "Slider Bars"
- Incluidas mas protecciones incorporadas
- Añadido una página adicional de la configuración del usuario incluyendo
    - Opción de mostrar el Día de la semana
    - Configurar la temperatura de arranque de los ventiladores
    - Añadido cuatro opciones adicionales definidas por el usuario para salvapantallas
- Añadido una función de Auto-Dimming (Atenuación automática de la intensidad de los LED) con opciones accesibles a través de la página de los ajustes generales.
    - Puedes escoger atenuar la intensidad de los LEDs cuando el disipador llega a una temperatura predefinida
    - Atenuar la intensidad de los LEDs en un porcentaje definido por el usuario.
    - Notificación visual en la pantalla principal si la función se utilizó de forma automática
- Se ha cambiado el aspecto de algunos botones y algunos retoques visuales.

Actualización esperada (Jarduino v.1.3)

- WiFi y la posibilidad de Aplicación para Android/iPhone

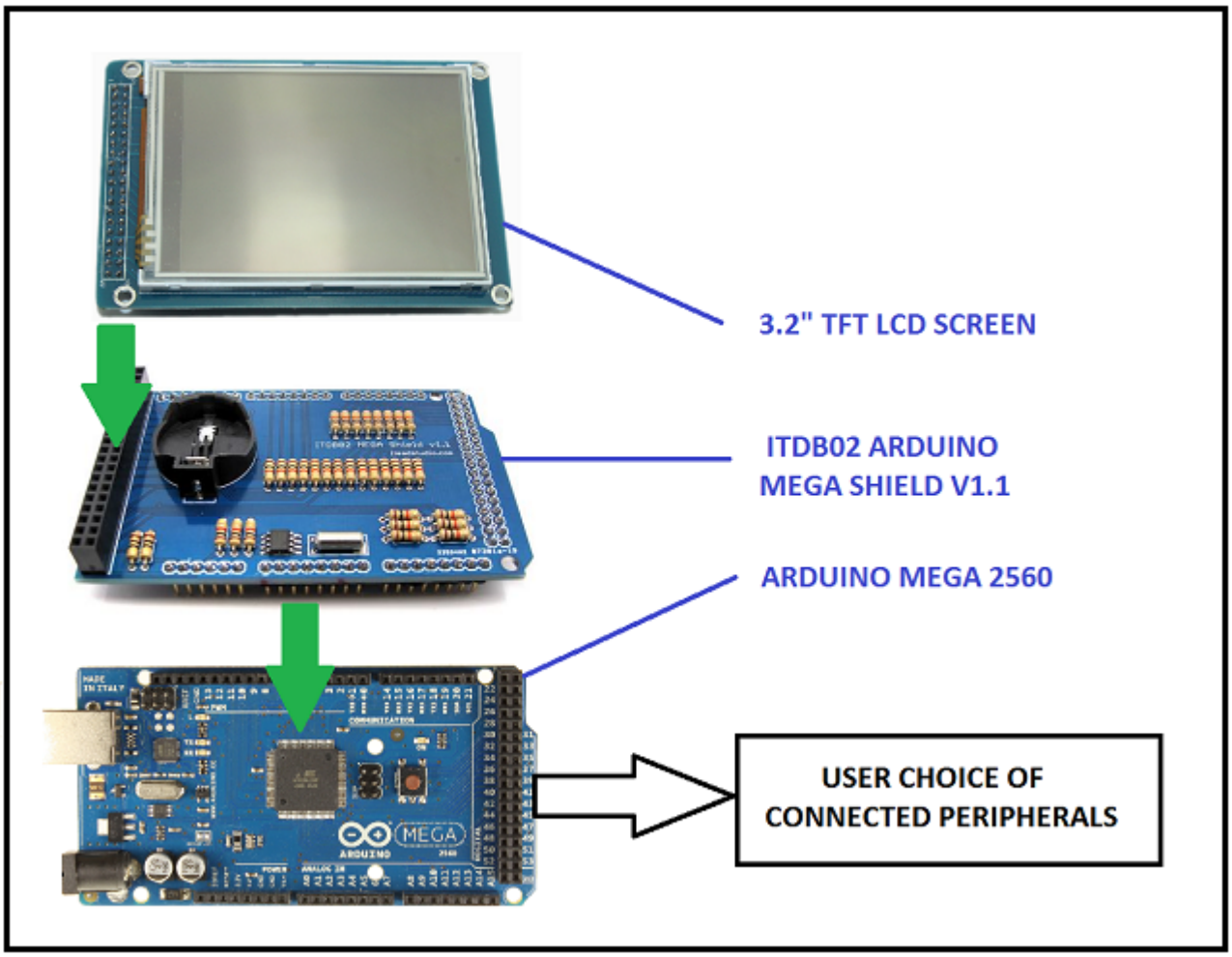
POSIBLES DESARROLOS ULTERIORES

- Interfaz de usuario mejorada (Pantalla principal & Atajos)
- Reemplazar la Librería ITDB02_Touch.h con la última versión (cuando esté disponible)
- Tiempo Aleatorio (Nubes, Tormentas, etc. junto con incrementar la acción de las olas)
- Tiempo Real (Reflejar las condiciones de tiempo reales de las ubicaciones seleccionadas por el usuario)
- Modification of Wave Maker to include Tidal Forces with Lunar Cycle
- Cambio Automático de Agua & Reposición (AWC & ATO)
- Incluir otros sensores:
  - pH
  - Salinity
  - Calcium
  - Orp
- Dosificación Automática
- Conexión inalámbrica de periféricos
- Video Vigilancia (echa un ojo a tu acuario desde cualquier ordenador o Smartphone)

*** POR FAVOR TENGA EN CUENTA – Yo no tengo una fecha límite para terminar cualquiera de las actualizaciones de la lista. Soy nuevo padre y estoy haciendo esto solo, así que por favor sea paciente!**

2. INSTRUCCIONES PARA CONSTRUIR EL JARDUINO

Cuando se trata de construir el Controlador de acuario Jarduino tienes varias opciones. En la introducción de este manual, esta presentada una distribución aproximada de mi diseño personal. A primera vista, puede parecer complicado, pero creo que esa impresión se debe al número de periféricos que he incluido en mi configuración. De hecho se puede utilizar el controlador Jarduino para controlar y supervisar el número de periféricos que deseas (por ejemplo - utilizarlo para controlar una sola serie de LEDs), o tantos como yo incluso muchos más. Cuando nos fijamos en los fundamentos, creo que se llega a la conclusión de que en realidad es bastante simple. Realmente, todo lo que se necesita para empezar son unos cuantos componentes clave, tal como se establece en el siguiente diagrama:



HARDWARE

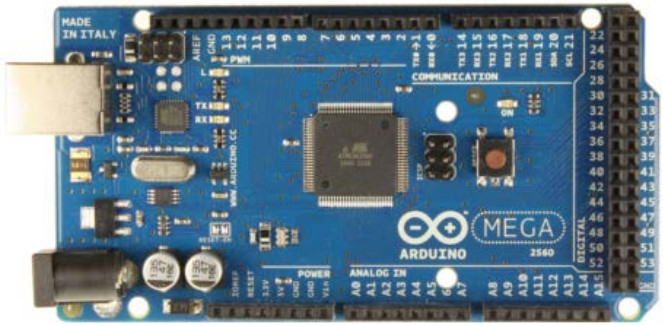
Arriba, hay tres componentes básicos que se presionan entre sí. En la base, hay un Arduino MEGA 2560. Se trata de una placa electrónica que es el "cerebro" detrás de las operaciones. Inmediatamente por encima del Arduino es un shield (escudo, pero vamos a utilizar el ingles) ITDB02 Arduino MEGA Shield v1.1 con un reloj de tiempo real incorporado (RTC). Este shield actúa como una interfaz entre el Arduino y la pantalla táctil LCD de 3,2 " que está representada en la parte superior del diagrama. Tenga en cuenta que existen algunas alternativas para cada uno de los componentes de hardware anteriores, que voy a tocar pronto. Pero, en esencia, esos tres componentes (o alternativas de los tres), son todo lo que se necesita para construir un controlador Jarduino. Lo único que queda después de haber construido el controlador Jarduino, es decidir qué componentes deseas controlar / supervisar, y luego conectarlos. Ok, admito que puede ser un poco difícil interconectar diferentes periféricos, pero una vez que aprendemos cómo conectar el primero, el siguiente que interconectamos será un poco más fácil, entonces, más fácil aún para un tercer componente, luego al final de tu construcción puedes sentarte y esperar a que las compañías de tecnología inicien una guerra de ofertas para tu empresa y tu recién adquirida experiencia en la electrónica para un contrato de trabajo lucrativo a largo plazo. ☺

LISTA DE COMPONENTES

En esta sección te presentamos una lista de compras. Si bien no cubre cada tuerca y tornillo necesario, sí que enumera todos los componentes principales y ofrece varias recomendaciones. Traté de incluir hipervínculos a algunos lugares de confianza donde he hecho una compra en el pasado. Si estás tratando de ahorrar un poco de dinero, muchas de estas piezas pueden ser adquiridas relativamente barato en eBay o "on line" en otros lugares. Puedes simplemente seguir cualquier enlace que proporciono, obtener las especificaciones de ese sitio, y tratar de encontrarlo más barato en otro lugar. Google es tu amigo cuando se trata de encontrar las piezas que necesitas. Consejo de compras: a veces es una buena idea comprar varias partes de un solo lugar, ya que su costo total puede ser más barato teniendo en cuenta los costes de envío.

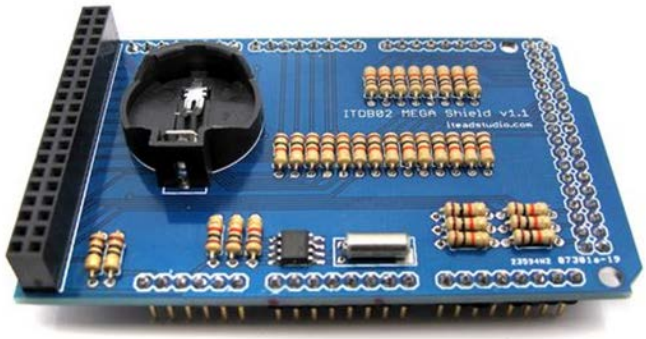
Componentes necesarios para el controlador Jarduino:

- **Arduino MEGA 2560** - eBay (presta atención a los votos del vendedor!)



NOTA: Hay varias versiones/variaciones del Arduino MEGA 2560. Cualquiera de ellas tendrá que ser compatible con Jarduino.

- **ITDB02 Arduino MEGA Shield Kit v1.1** – iTeadStudio.com / iMall (El enlace que he proporcionado es de un kit que tienes que montarlo (soldar). Hay algunos lugares que venden este shield completamente montado)



NOTA: Hay una nueva versión disponible de este shield: ITDB02 Arduino MEGA Shield v2.0. Esta nueva versión no tiene incorporado un reloj de tiempo real (RTC), así que si decides comprar esa versión, tendrás que añadir un circuito RTC con el fin de hacerlo compatible (que yo te mostraré cómo hacerlo en la sección de montaje siguiente). También hay unos pocos shields que hay por ahí de otros fabricantes. Ninguno de estos shields que he visto en la actualidad se ha construido con RTC, que el Jarduino requiere. Y por alguna razón, la v1.1 ITDB02 del shield es menos cara. Yo sólo recomendaría conseguir una versión distinta de los shield ITDB02 v1.1 en el caso de que no puedes encontrar los Shield ITDB02 v1.1.

En la foto de la parte izquierda un ITDB02 MEGA Shield v2.0, en la derecha la versión de "ELECFreaks." Ninguna de las dos tiene RTC!

- **3.2" TFT LCD** – iTeadStudio.com / iMall (He utilizado este de iTead solo porque tuve la oportunidad de ahorrar en el envio por comprar esto y el ITDB02 Shield Kit v1.1 juntos)

NOTA: A continuación se muestra una lista truncada de módulos LCD TFT compatibles que fue compilada originalmente por Henning Karlsen, quien escribió el ITDB02_graph16 y las Bibliotecas(Library) UTFT Arduino. Yo uso la palabra "compatible" aquí libremente. El Jarduino fue escrito expresamente para un tamaño de pantalla de 320 x 240 píxeles. Por lo tanto, mientras que los módulos de pantalla de abajo podrían ser "compatibles", no necesariamente se muestre correctamente si la pantalla no es exactamente 320 x240. Por ejemplo, si deseas utilizar la de 5" ITDB02-5.0, que es de 720 x 480 píxeles, el Jarduino va a llenar sólo una parte de la pantalla (los primeros 320 píxeles a través, seguido de una barra de 720-320 = 400 pixeles de ancho de color negro, y los primeros 480-240 = 240 pixeles bajando, seguido de una barra de 240 píxeles de ancho de color negro debajo de la misma). Si realmente quieres usar esa pantalla en particular (y llenarlo correctamente), o cualquier otro que no es 320x240, entonces tendrás que pasar por el Sketch (programa) Jarduino y cambiar todas las líneas en el código que tienen que ver con el formato de pantalla. Esto no es imposible, pero ten en cuenta que el programa tiene casi 6000 líneas.

Los siguientes módulos LCD han sido probados y funcionan con ITDB02_graph16 Library. Esta es más rápida que la biblioteca UTFT pero menos compatible. Otros módulos tendrían que funcionar siempre que tengan uno de los controladores compatibles. Yo recomiendo utilizar uno de los siguientes módulos de 3.2" (subrayados en amarillo) debido al tamaño de la pantalla (320 x 240):

Proveedor: ITead Studio

Module	Model for ITDB02()	Controller	Notes
ITDB02-3.2	ITDB32	HX8347-A	
ITDB02-3.2WC	ITDB32WC	ILI9327	
ITDB02-3.2S	ITDB32S	SSD1289	

Proveedor: ElecFreaks

Module	Model for ITDB02()	Controller	Notes
TFT01-2.4	TFT01_24	ILI9325D	
TFT01-3.2	TFT01_32	SSD1289	
TFT01_3.2W	TFT01_32W	ILI9327	

Los siguientes módulos LCD han sido probados y funcionan con UTFT Library. Esta es más LENTA que la biblioteca ITDB02_graph16 pero MÁS compatible. Otros módulos tendrían que funcionar siempre que tengan uno de los controladores compatibles.

Proveedor: ITead Studio

Module	Model for ITDB02()	Controller	Notes
ITDB02-1.8SP	ITDB18SP	ST7735	
ITDB02-2.2SP	ITDB22SP	HX8340-B(N)	

ITDB02-2.2	ITDB22	HX8340-B(T)	
ITDB02-2.4	ITDB24	ILI9325C	Retired
ITDB02-2.4D	ITDB24D	ILI9325D	
ITDB02-2.4DWOT	ITDB24DWOT	ILI9325D	
ITDB02-2.4E	ITDB24E_8	S6D1121	8bit mode
ITDB02-2.4E	ITDB24E_16	S6D1121	16bit mode
ITDB02-2.5H	ITDB25H	S1D19122	
ITDB02-2.8	ITDB28	ILI9325D	Retired
ITDB02-3.2	ITDB32	HX8347-A	Retired
ITDB02-3.2WC	ITDB32WC	ILI9327	
ITDB02-3.2S	ITDB32S	SSD1289	
ITDB02-3.2WD	ITDB32WD	HX8352	
ITDB02-4.3	SSD1963	SSD1963_480	
ITDB02-5.0	SSD1963	SSD1963_800	

Módulos compatibles con UTFT Library (continuación):

Proveedor: ElecFreaks

Module	Model for ITDB02()	Controller	Notes
TFT01-2.4	TFT01_24_8	ILI9325D	8bit model
TFT01-2.4	TFT01_24_16	ILI9325D	16bit model
TFT01-3.2	TFT01_32	SSD1289	
TFT01_3.2W	TFT01_32W	ILI9327	Retired
TFT01_3.2WD	TFT01_32WD	HX8352	

Proveedor: NKC Electronics

Module	Model for ITDB02()	Controller	Notes
RGB LCD 65K color module	LPH9135	PCF8833	

Si no estás seguro que módulo LCD tienes, puedes intentar los códigos siguientes para los controladores compatibles con la biblioteca UTFT Library:

Controller	Model for UTFT()	Supported mode		
		8bit	16bit	Serial
HX8340-B(N)	HX8340B_S			✓
HX8340-B(T)	HX8340B_8	✓		
HX8347-A	HX8347A		✓	
HX8352-A	HX8352A		✓	
ILI9325C	ILI9325C	✓		
ILI9325D	ILI9325D_8	✓		
ILI9325D	ILI9325D_16		✓	
ILI9327	ILI9327		✓	
PCF8833	PCF8833			✓
S1D19122	S1D19122		✓	
S6D1121	S6D1121_8	✓		
S6D1121	S6D1121_16		✓	
SSD1289	SSD1289		✓	
SSD1963	SSD1963_480		✓	
SSD1963	SSD1963_800		✓	
ST7735	ST7735			✓

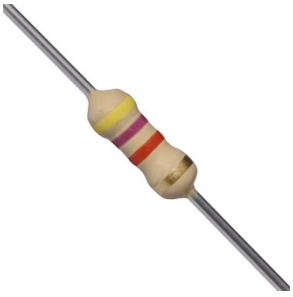
COMPONENTES RECOMENDADOS para el controlador Jarduino:

❖ COMPONENTES PARA EL SENSOR DE TEMPERATURA

- **Sensor de temperatura DS18B20** – eBay (Algunos vendedores ofrecen estos el "lotes". Recientemente compre 10 por unos 7 USD. Incluso hay vendedores que los venden conectados a un cable y sellados en acero inoxidable como el de la foto de la derecha).

NOTA: El código del Jarduino está escrito para utilizar estos sensores de temperatura en particular. He utilizado tres de estos en mi controlador. Puedes utilizar el mismo número, más, o menos en el tuyo. Con unas pocas modificaciones en el código y la conexión de los pines, otros sensores de temperatura (como los analógicos LM35) pueden ser sustituidos. Con el LM35 sin embargo se limitarían el número de pines analógicos disponibles.

- **Resistencias 4.7 kΩ** – Cualquier tienda local de electrónica o "Online"



NOTA: Necesario para el sensor de temperatura DS18B20; Necesitas solo una resistencia, no importa cuántos DS18B20's decide utilizar.

❖ MODULO RELOJ DE TIEMPO REAL

***** *NOTA: Esto lo necesitas SOLO si utilizas otro shield que no es el ITDB02 Arduino MEGA v1.1.*

- **Módulo RTC DS1307**– eBay o cualquier tienda online (He comprador este modelo en particular en eBay por $2.85 USD. Tenía envio gratuito y viene con batería.)

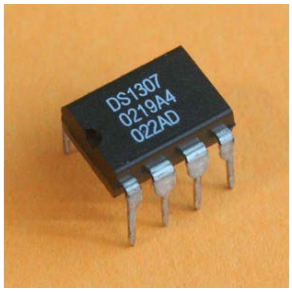
TOP VIEW

BOTTOM VIEW

❖ **COMPONENTES RELOJ DE TIEMPO REAL (RTC)**

** NOTA: Esto lo necesitas SOLO si utilizas otro shield que no es el ITDB02 Arduino MEGA v1.1Puedes utilizar un módulo RTC como el de arriba, o puedes construir uno. Abajo hay un listado de los componentes necesarios para construir uno tú mismo.*

☐ **DS1307 RTC Chip** – eBay o cualquier tienda online



NOTA: Este es un chip de Reloj en Tiempo Real serial.

☐ **Cristal 32.768kHz** – eBay o cualquier tienda online

NOTA: Esto es la fuente del reloj para el chip RTC

☐ **Resistencias 10 kΩ** - Cualquier tienda local de electrónica o “Online”

NOTA: Vas a necesitar .dos de estas para el circuito del reloj RTC

☐ **Batería modelo CR2032** – En cualquier tienda que vende baterías

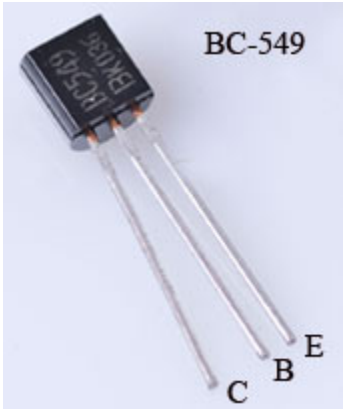
NOTA: Esta es la batería de reserva para RTC, y hay varias marcas disponibles.

- **Porta pilas para CR2032**– eBay o cualquier tienda online

NOTA: Es un soporte pequeño para la batería del RTC.

❖ VARIOS COMPONENTES

- **Transistores NPN BC549C** – Cualquier tienda local de electrónica o "Online"

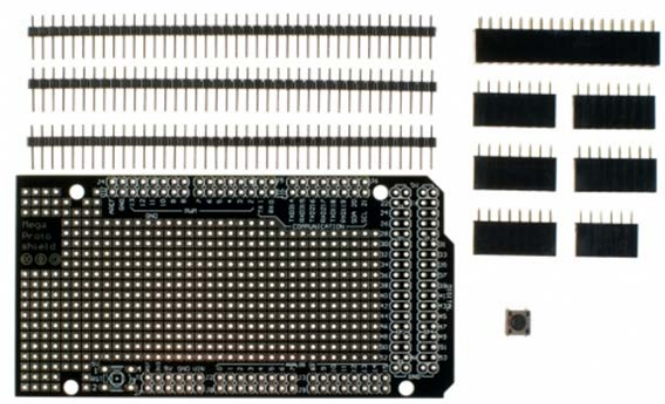


NOTA: Utilizado en varios circuitos para controlar algunos periféricos (ventiladores, comedero automático, alarmas). Compra uno de estos para cada periférico que planeas implementar. Pueden ser sustituidos con transistores que tienen características similares como por ej. El BD137.

- **Resistencias 1 kΩ** – Cualquier tienda local de electrónica o "Online"

NOTA: Utilizado en varios circuitos para controlar algunos periféricos (ventiladores, comedero automático, alarmas). Compra uno de estos para cada periférico excepto para LEDs. Utilice una cada vez que utiliza uno de los transistores de arriba.

- **Placa de prototipo para Arduino MEGA**– DFRobot (Utilicé la de DFRobot porque ofrece la mayor superficie de prototyping en comparación con otras que he visto)



NOTA: Se utiliza para la interconexión de todos los accesorios a la Arduino. Esto no es en absoluto la única opción, pero hace el trabajo, y proporciona un montón de espacio para poner todos los circuitos. He utilizado dos de estos en mi controlador. Alternativas potenciales incluyen: conexiones directas con la parte inferior Arduino, las conexiones con el shield ITDB02, tarjeta perforada, otros protoshields, y los shields de tornillo para nombrar unas pocas. Si decide utilizar las placas de prototipos, yo recomiendo usar las destinadas al Arduino MEGA. Comparado con otras que he visto.

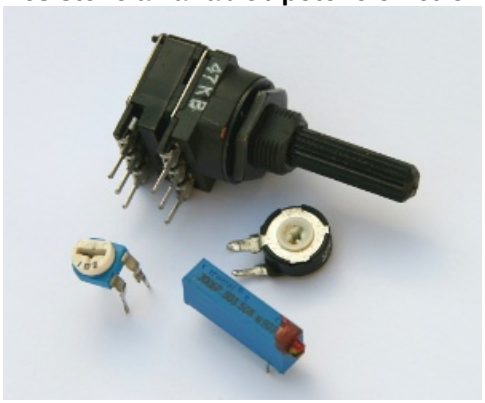
COMPONENTES OPCIONALES para el controlador Jarduino:

❖ COMPONENTES PARA ALARMA AUDIO

- **Timbre / Buzzer pasivo de 5V**– Cualquier tienda local de electrónica o "Online"

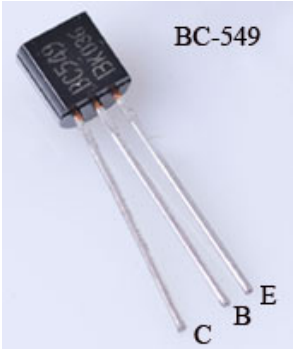
NOTA: Utiliza uno de estos si deseas una alarma audible. El Jarduino hará sonar una alarma si la temperatura del agua sube o baja por encima de un valor definido por el usuario. Esta misma puede ser utilizada para que suenen otras alarmas definidas por el usuario con unas pocas líneas de código.

- **Resistencia variable / potenciómetro** – Cualquier tienda local de electrónica o "Online"



NOTA: Si deseas una alarma, añade uno de estos para controlar el volumen. Un potenciómetro con un valor de máx. 10k debería ser suficiente. Hay muchos tipos y modelos de donde escoger.

- **Transistor NPN BC549C & Resistencia1 kΩ** – Cualquier tienda local de electrónica o “Online”



NOTA: Ambos figuraban en Varios componentes. Simplemente quiero recordar que forman parte del circuito de alarma.
.

❖ COMPONENTES PARA EL GENERADOR DE OLAS

- **Enchufe / Caja para enchufe / Tapa enchufe** – Tiendas locales de electricidad

NOTA: En la foto de la izquierda es un dúplex de 15 amp que normalmente se encuentran en los hogares de Estados Unidos. Los otros elementos representados son algunas de las opciones disponibles para revestir la salida (salida simple, caja de salida y placa de pared). Sustituir estos elementos por cualquier tipo de enchufe de su país. No está limitado al uso de estos tipos. Puede incluso modificar alargadores de tira utilizando las mismas partes y circuitos.

- **Solid State Relay KYOTTO KB20C06A (SSR)** – eBay o cualquier tienda online

NOTA: Estos relés en particular tienen una corriente de fuga muy baja, por lo que no son ideales para controlar las bombas de movimiento. De todos modos los he utilizado en mi controlador, y trabajan muy bien con las bombas Hydor Koralia. Elegí los SSR, porque

son completamente silenciosos. Pueden fácilmente ser sustituidos los relés mecánicos con características similares. Son mucho menos costosos, pero hacen perceptibles chasquidos cuando se activan.

- **Transistor Darlington ULN2803A**– eBay o cualquier tienda online

NOTA: Este transistor puede controlar hasta 8 tomas de corriente.

- **Zócalo de 18-Pin DIP IC** – eBay o cualquier tienda online

NOTA: Esto es completamente opcional. Es muy útil si alguna vez tienes que reemplazar el ULN2803A.

- **Condensadores X2 (0.22µF / 270VAC)** – eBay o cualquier tienda online

NOTA: Otro componente opcional. Lo he utilizado para cumplir la directiva EMC. Pero esto es DIY así que no importa mucho. Si vas a utilizarlos, hay que poner uno en cada toma de corriente.

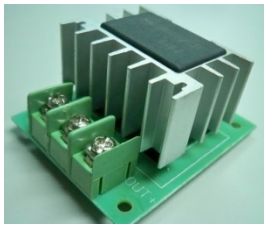
¿Quieres controlar otros componentes de CA como un calentador, enfriador, Luces T5, reactor de calcio, o algo más? A continuación, puede utilizar exactamente las mismas piezas que se enumeran más arriba para el Generador de Olas para controlarlos. Sólo tienes que añadir una toma de corriente adicional por cada periférico de CA. Además del Generador de Olas, el código del Jarduino ya se ha escrito para controlar un calentador y un enfriador. Otros periféricos de CA son bastante fáciles de codificar y muy fácil de controlar.

❖ MAS COMPONENTES VARIOS

- **Diodos IN4001** – Cualquier tienda local de electrónica o "Online"

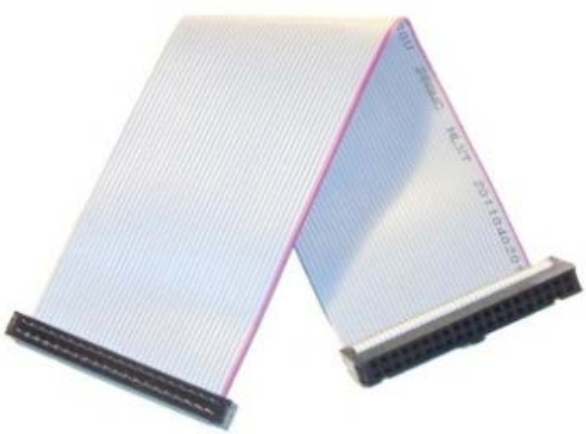
NOTA: He utilizado este diodo en particular para la protección de los ventiladores. Si no quieres usarlos puedes prescindir de ellos. También pueden ser sustituidos con diodos que tienen especificaciones similares. Lo que es más cómodo.

- **HRD Series DC Step Down Converters (Adjustable 36V-5V)** – eBay



NOTA: He utilizado 3 de ellos para bajar el voltaje de la fuente de alimentación principal a cualquier voltaje que necesitaba para los distintos periféricos. En particular, se utiliza para reducir los 36 voltios de la fuente de alimentación a 11 V para mi Arduino, 12 V para mis ventiladores, y 3 V para mi comedero automático (he añadido unos cuantos diodos en serie para bajar el voltaje de 3 V, ya que el convertidor sólo bajó la tensión a 5v). He utilizado estos, así que no tenía necesidad de suministros adicionales de energía para mis periféricos. Todo está alimentado de una fuente de alimentación principal.

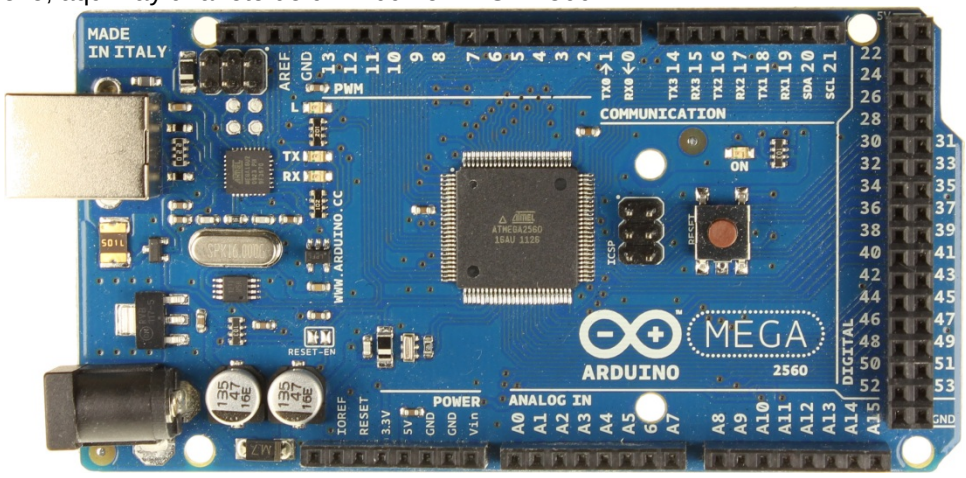
- **Cable Ribbon 40-Pin IDE (Macho-Hembra)** – Cualquier tienda local de electrónica o “Online”



NOTA: Puedes usar esto para alejar la pantalla LCD del shield ITDB02 Arduino MEGA. Asegúrate de que tienes un cable 40-pin IDE que tiene un extremo que es hembra, y el macho al otro extremo. Además, asegúrate de que no hay "bloques" en los conectores. Estos conectores se usan comúnmente para conectar CD / DVD-ROM a una placa base. NO SE SABE la longitud de cable que se puede utilizar. He probado con éxito hasta 30 cm (1 pie) sin ningún problema.

MONTAJE DEL JARDUINO

Hay un par de cosas que quiero asegurarme que entiendes antes de empezar a montar el controlador de acuario. En primer lugar, vamos a hablar un poco sobre el Arduino MEGA 2560. No voy a aburrirte con todas las especificaciones y características. Sólo quiero asegurarme de que entiendes dónde "enchufar" las cosas cuando yo le indicará que lo haga. Así que de nuevo, aquí hay una foto de un Arduino MEGA 2560:



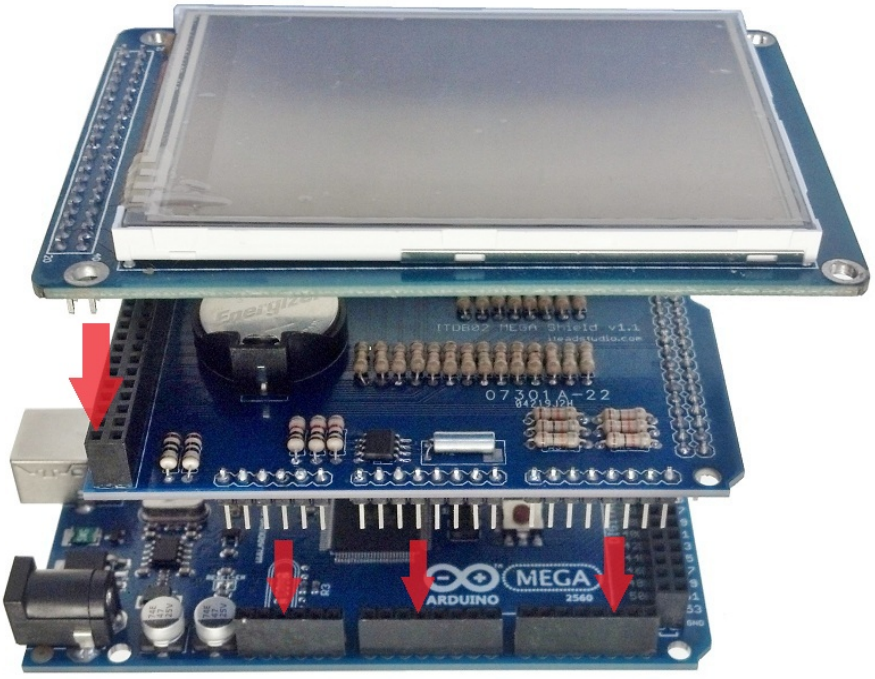
En la placa, debería ser capaz de ver que cada pin está claramente identificado con un número o una breve descripción de lo que el pin es. Por otra parte, los pines se agrupan en categorías "PWM" para los pines 2-13, "digitales" a la derecha para los pines 22-53, etc En el código del programa Jarduino, definí en que pin se conecta cada periférico. Se trata de una asignación de pines. Puedes cambiar algunas de las asignaciones de pines si

quieres, pero muchos de ellos no se pueden cambiar. En particular, los pines que se utilizan por la pantalla de TFT LCD no se pueden cambiar. Así es como los pines están actualmente definidos en el programa:

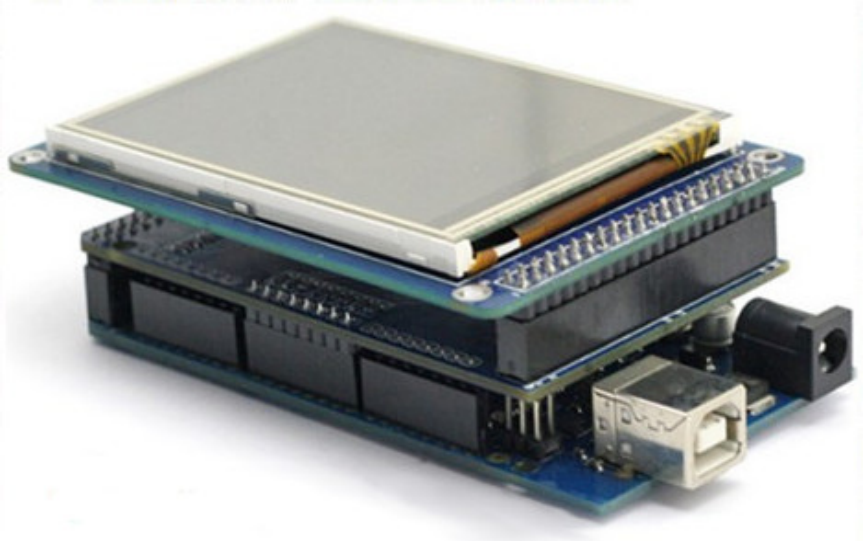
ASIGNACIÓN DE PINES

Controller Peripheral / Component	Pin Assignment(s)
TFT LCD Shield	2, 3, 4, 5, 6 38, 39, 40, 41
ITDB02 Shield	22-37
LEDs: Sump	7
LEDs: Regular Blue (AZUL)	8
LEDs: White (BLANCO)	9
LEDs: Royal Blue (AZUL ROYAL)	10
LEDs: Red (ROJO)	11
LEDs: Ultraviolet (UV)	12
LEDs: Moonlight(LUZ DE LUNA)	13
Reloj de tiempo real (DS1307 RTC)	20, 21
Generador de olas – bomba uno	42
Generador de olas – bomba dos	43
Ventiladores: Para Pantalla(PWM)	44
Ventiladores: Para luz sump (PWM)	45
Ventiladores (ON/OFF): Para Pantalla	47
Ventiladores (ON/OFF): Para luz sump	48
Alarma audio para temperatura	49
Comedero Automático	50
Sensores de temperatura (DS18B20)	51
Calentador	52
Enfriador	53

Si has estado prestando atención hasta el momento, ya debes saber cómo armar los componentes principales del Jarduino. Pero para estar seguros, con el Arduino en la base "prensa" en el ITDB02 Arduino MEGA Shield, luego en la parte superior "prensa" la pantalla LCD 3,2 ", así:



Con todos encajonados juntos, debería tener este aspecto:

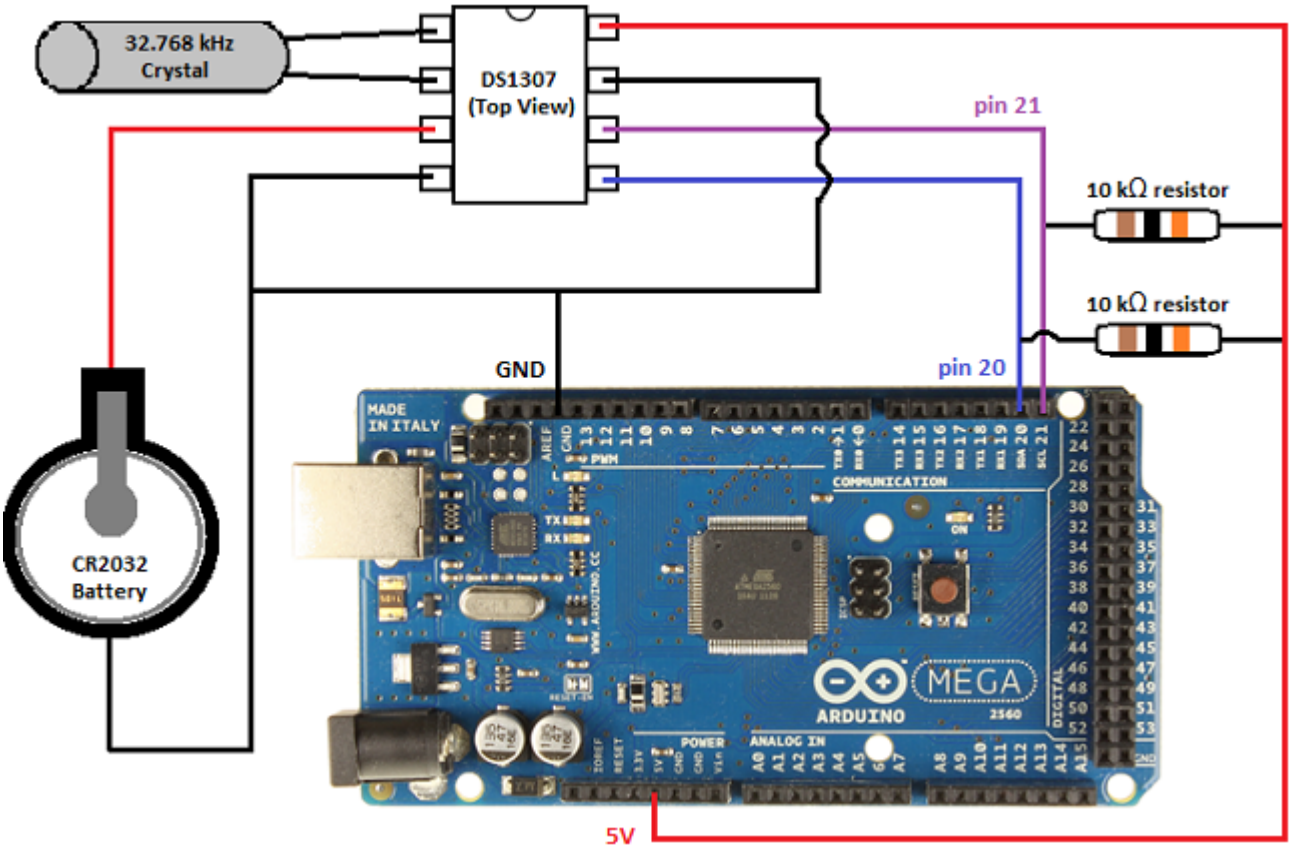


¡Qué fácil era eso! En este punto puedes estar ansioso de ver el Jarduino en funcionamiento en el controlador. La buena noticia es que puedes cargar el programa en estos momentos. Consulta CARGANDO EL CODIGO DEL JARDUINO EN EL ARDUINO abajo para aprender cómo hacerlo. Antes de hacerlo, sin embargo, hay algunas cosas que puedes tener en cuenta. En primer lugar, si no has utilizado un ITDB02 v1.1 Arduino MEGA Shield Expansion LCD con el RTC integrado, tendrás que conectar un circuito RTC. Además, es posible que desees conectar el sensor (sensores) de temperatura DS18B20. Dejando de lado un RTC y Sensores de temperatura no evitará que se cargue el programa. Simplemente te dejará con algunos errores en la pantalla hasta que consigues conectarlos correctamente. Así que si quieres cargar el Jarduino, adelante, hazlo ahora. Más tarde, una vez que los sensores de temperatura (RTC y de ser necesario) conectados, sólo tienes que volver a cargar el programa del Jarduino.

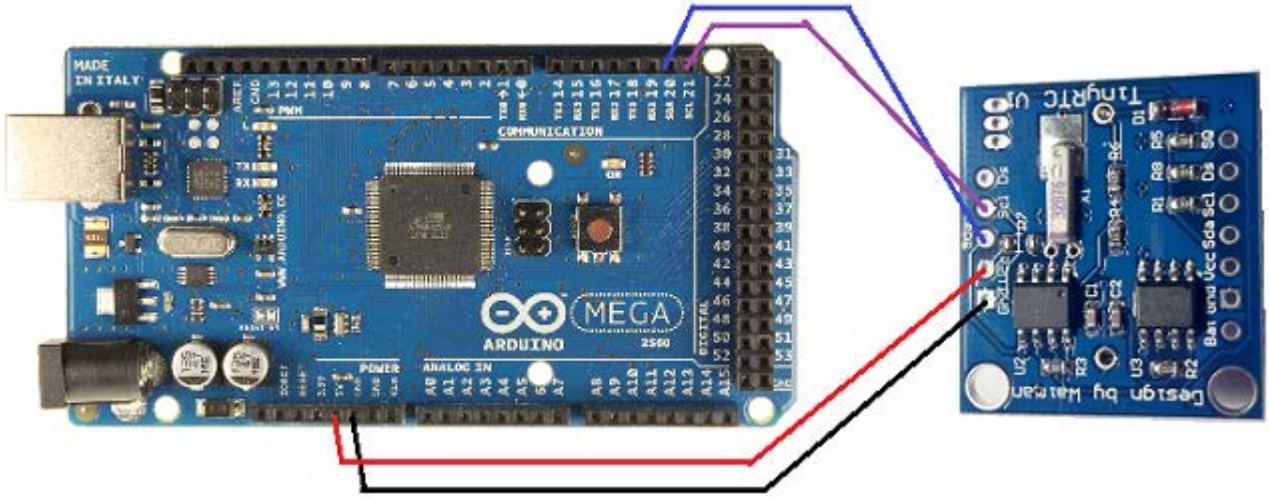
CONECCIÓN DEL RTC

Salta este paso si estás usando un ITDB02 v1.1 shield Arduino MEGA con el RTC integrado.

Si todavía estás leyendo esto, entonces por alguna razón, no tienes un ITDB02 v1.1 shield Arduino MEGA con el RTC. Oh, bueno, eso sólo significa que tendrás que añadir tu propio RTC. Aquí hay un dibujo del circuito RTC:



Así que la construcción de este circuito no es demasiado complicada. Simplemente cablea y suelda todo como lo ésta indicado anteriormente. Puedes ponerlo en una placa perforada para prototipos, o directamente soldarla a la parte inferior de la placa Arduino (o soldarla en la parte superior del shield de expansión de la pantalla LCD. Tienes mucha libertad aquí. Siempre y cuando lo tienes conectado correctamente debería funcionar. Incluso más fácil de conectar sería un módulo pre montado RTC (sólo 4 cables).

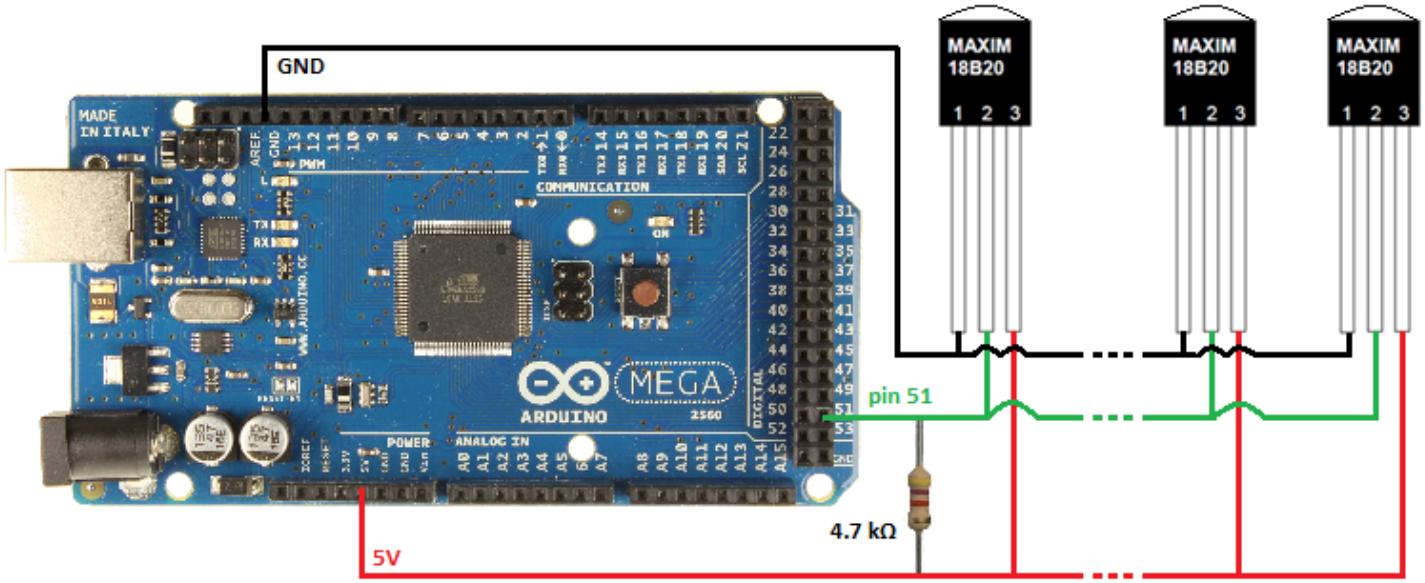


NOTA: Dependiendo del módulo RTC, las posiciones de conexión pueden variar ligeramente.

CONECCIÓN DE LOS SENSORES DE TEMPERATURA

La primera vez que vas a descargar el Jarduino, vas a encontrar el código en el mismo estado en que lo dejé cuando lo he subido a mi Arduino. Por lo tanto, es mi configuración específica. Aunque no es necesario hacer cambios importantes en el código, tendrás que determinar el número serial específico para cada uno de los sensores de

temperatura. Una vez que obtengas los números de serie, tendrás que entrar en el código Jarduino. En la sección Software abajo, te enseñaré hacer ambas cosas. Pero en primer lugar, con el fin de obtener los números de serie, tendrás que seguir este diagrama para conectar los sensores de temperatura con el Arduino:

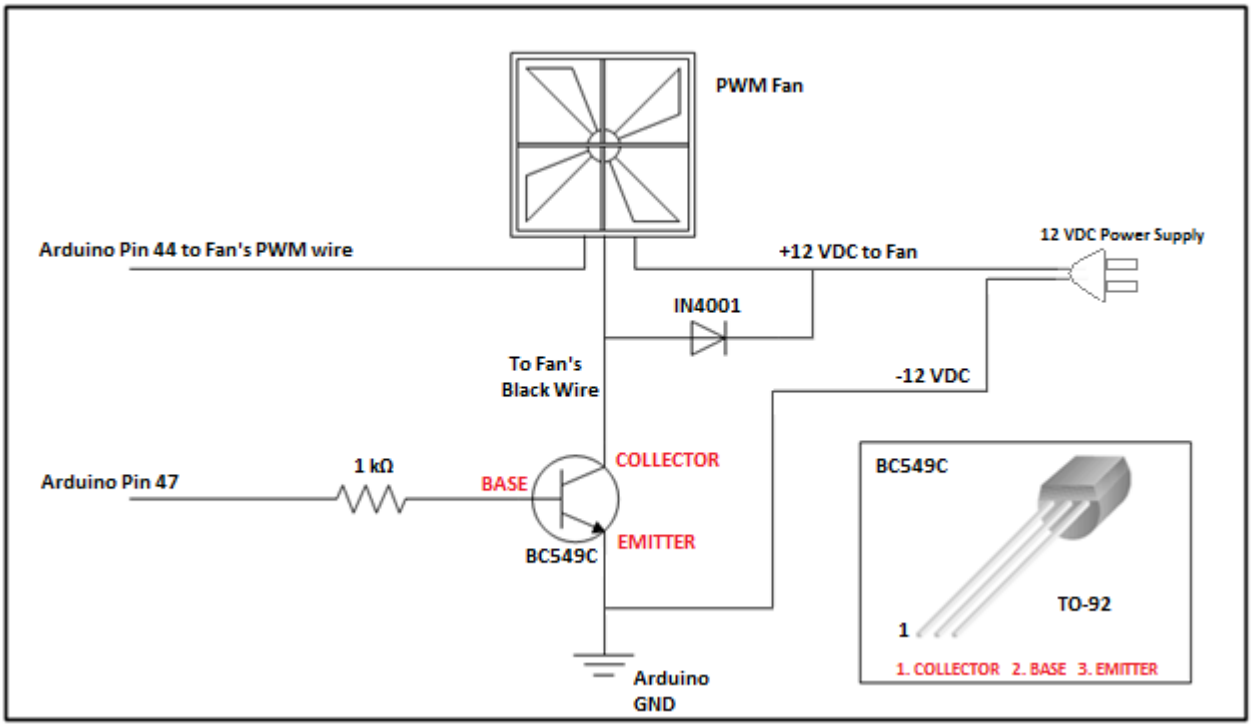


Observas los dos sensores en el extremo derecho. He interrumpido los cables para indicar que se pueden conectar más de tres sensores a la placa Arduino. Sin embargo, sólo hay una resistencia de 4,7 k. No importa cuántos sensores decides conectar, se utiliza sólo una de estas resistencias. Creo que es más conveniente para conectar la resistencia en algún lugar cercano a la placa Arduino, y desde ese punto partir con los cables que conectan los sensores de temperatura hasta el lugar que deseas supervisar. Creo que los cables de teléfono funcionan bien aquí, ya que tienen por lo menos tres cables en su interior. Asegúrate de impermeabilizar bien cualquier sensor que planeas sumergir en el agua. Como ya he dicho, varios fabricantes venden estos encapsulados en una carcasa de acero inoxidable impermeable. Cuestan un poco más, pero te puedes ahorrar algo de trabajo.

CONECCIÓN DE LOS VENTILADORES

El Jarduino proporciona una refrigeración dinámica del disipador de los LED a través del uso de ventiladores PWM - "Pulse Width Modulated" (PWM). Cuanto más se calienta el disipador (temperatura medida por uno de los sensores de

temperatura DS18B20), más rápido girará el ventilador. Si el disipador es lo suficiente frío, se corte la alimentación del ventilador para ahorrar energía. Los ventiladores son muy silenciosos, funcionando a una frecuencia de 25 kHz, lo que está fuera de nuestro rango de audición. Así es como yo les he conectado a la placa Arduino. Estoy asumiendo que ahora es probable que entiendas un esquema más simbólico.



CONECCION DE LOS LED

Como probablemente ya sabes, hay muchas maneras diferentes en que puedes conectar tus LED, qué tipo de tipo LED puedes utilizar, y que tipo de driver (fuente). Para complicar las cosas, hay que preguntarse qué colores debes utilizar, cuántos de cada uno, de que potencia. Yo ni siquiera voy a tratar de escribir una guía sobre esto, ya que hay demasiadas variaciones posibles que dependen demasiado de la preferencia del usuario. Como yo siento que hay un montón de buena información disponible en ayudarte a tomar una decisión sobre lo que es mejor para tus necesidades, simplemente asumiremos que ya has tomado tu decisión en lo que respecta a los LEDs que deseas utilizar y, que tipo de fuente vas a utilizar para encenderlos.

Las principales cosas que debes saber acerca de la Jarduino y los LED que controla son las siguientes:

- El Jarduino utiliza señales PWM para controlar (dimmear) la intensidad de los LEDs, por lo que es necesario asegurarse de que los controladores (drivers, fuentes) de LED son capaces de ser dimmeados a través de señales PWM. No importa si el driver requiere una fuente de alimentación separada o es un todo-en-uno; Jarduino puede controlarlos mientras se pueden dimmear mediante PWM.
- Si deseas controlar una pantalla LED comercial, los driver o las fuentes de esta deben ser controlados a través de PWM. Si no es así, siempre se puedes reemplazar los drivers o las fuentes con unos que son dimmeables mediante PWM.
- En la actualidad, el Jarduino puede controlar hasta 5 canales separados de LEDs. Puedes dedicar un color diferente para cada canal. También puedes agregar el número de LEDs que quieras a cada canal (siempre y cuando puedes proporcionar la energía necesaria para encenderlos).
- La fase de la Luna tiene su propio canal, por lo que tendrá LEDs dedicados junto con su propio driver LED. Sin embargo no deberías tener demasiados LEDs como luz de luna. (Yo uso dos CREE XP-E Royal Blue en mis 44 galones y es más de lo suficiente).

Los únicos cables que necesitas para conectar desde el Jarduino a los controladores (drivers, fuentes) son una masa (GND) y los pines PWM para cada color respectivo. Por supuesto, tendrás que alimentar el LED, pero por el bien de este manual, voy a suponer que ya sabe cómo o has descubierto la manera en otro lugar. Sin ningún tipo de modificación en el código Jarduino, estos son los colores que el Jarduino controla junto con los respectivos números de pin de Arduino:

- Pin 7: SUMIDERO / Refugio (blanco)
- Pin 8: Azul Regular
- Pin 9: Blanco
- Pin 10: Royal Blue
- Pin 11: Rojo
- Pin 12: UV (Ultravioleta)
- Pin 13: Luz de Luna

Nota: Puedes cambiar los colores y las asignaciones de los pines en el esquema Jarduino para lo que quiera. Esos son simplemente los de mi configuración.

3. INSTRUCCIONES DE SOFTWARE

SOFTWARE

El entorno de desarrollo Arduino (IDE) se puede obtener de aquí: http://arduino.cc/en/Main/Software

Todas las versiones del Código Jarduino se puede obtener de aquí: http://code.google.com/p/jarduino-aquarium-controller/

Diversas librerías de Henning Karlsen: http://www.henningkarlsen.com/electronics/library.php

Aquí hay algunos enlaces a unos pocos archivadores de ficheros: WinZip, WinRAR, 7-Zip (no tienes que usar estos, pero vas a necesitar un archivador de ficheros)

VERSIONES JARDUINO

Jarduino v1.1 – Abril 8, 2012
Este archivo contiene una versión modificada del Medio Arduino 0022. Es modificada para corregir un error que aparece al intentar cargar sketches grandes. También se incluye el Jarduino Aquarium Controller v1.1, junto con todas las librerías necesarias para cargar con éxito el programa para el Arduino MEGA 2560. El parche Arduino IDE 0022 es el único IDE que se sabe de cargar con éxito el Jarduino v1.1.

Jarduino v1.2 (beta1) – Mayo 19, 2012
Este archivo contiene la versión beta del Jarduino Aquarium Controller v1.2, junto con todas las librerías necesarias para cargar con éxito el programa para el Arduino MEGA 2560. Para obtener una lista completa de características, consulte la página 3 de este documento. También se incluye en este archivo una versión modificada de Arduino 1.0. Es modificada para corregir un error que aparece al intentar cargar sketches grandes. El Arduino IDE 1.0 parcheado subirá con éxito el Jarduino v1.2 beta, así como todas las versiones posteriores de la Jarduino. Arduino IDE 1.0.1 (disponible en el sitio web de Arduino) debe cargar con éxito el Jarduino v1.2 beta y todas las versiones posteriores sin la necesidad de usar mi versión parcheada de la IDE. Ha habido algunos informes de no ser capaz de utilizar Arduino 1.0.1. Si experimenta el mismo problema, debe volver a utilizar la versión parcheada de Arduino IDE 1.0 incluida en esta versión.

Jarduino v1.2 (beta1 with ITDB02_Graph16) – May 24, 2012
Este archivo contiene sólo un cambio de la versión anterior del Jarduino Aquarium Controller v1.2 (beta 1). Sustituye la nueva librería UTFT con la antigua librería ITDB02_Graph16. La librería UTFT tiene más compatibilidad, mientras que la librería ITDB02_Graph16 es más rápida. Más sobre la que uno debe elegir, será cubierto en este documento más adelante.

Jarduino v1.2 (beta2) – June 28, 2012
Este archivo contiene una versión modificada del controlador Jarduino Aquarium v1.2, versión beta. Esta versión de v1.2 Beta corrige el Auto-Dimming de los LED cuando alcanzan una temperatura definida por el usuario. También algunos retoques y correcciones han sido hechos.

CARGANDO EL PROGRAMA JARDUINO EN EL ARDUINO

Este tutorial está dirigido a los usuarios de Windows, pero otros sistemas operativos seguirán de cerca el mismo procedimiento. Instrucciones más detalladas sobre cómo conectar el Arduino al ordenador se encuentran disponibles en la página web del Arduino.

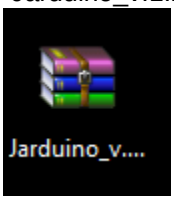
PASO 1: Instalar el archivador de ficheros

Aunque es probable que puedes tener esto en el equipo ya, lo primero que se necesita es software de archivado. Los ejemplos incluyen, pero no se limitan a: WinZip, WinRAR y 7-Zip. Algunos de estos programas cuestan dinero, pero suelen venir con una versión de prueba gratuita. Otros 7-Zip y muchos son gratuitos. Puedes usar cualquier archivador que mejor se adapte a tus necesidades. Necesitarás un software de archivado para "descomprimir" los archivos de mi página de descarga, así como los archivos de Arduino.

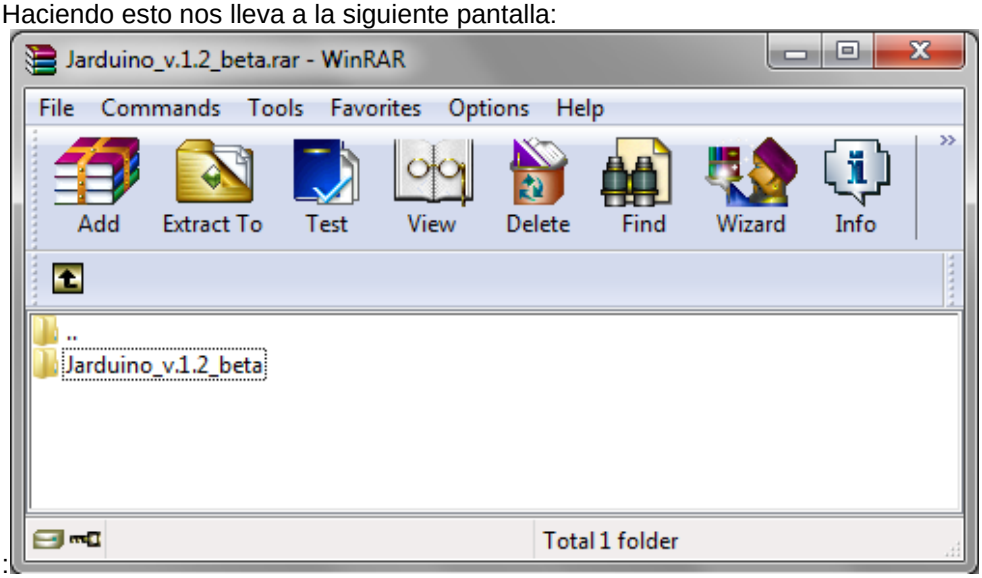
PASO 2: Descargar e instalar el software de desarrollo Arduino

Asumiendo que obtuviste una copia de la IDE Arduino que necesitas, lo siguiente que hay que hacer es "instalarla" en tu ordenador. Para Jarduino v1.1, usar mi versión parcheada de Arduino 0022. Para Jarduino v1.2 y posterior, puedes instalar cualquier versión parcheada de mi Arduino 1,0, o utilizar la versión oficial de Arduino 1.0.1 (disponible en la página web del Arduino).

Para este ejemplo, voy a descargar Jarduino v1.2 (beta1) porque tiene mi versión parcheada de Arduino 1.0 junto con todas las librerías que necesito y por supuesto el programa del Jarduino. Escogí para guardarlo en mi escritorio. Una vez que el archivo termine de descargar, haz "doble clic" en el archivo de la carpeta "Jarduino_v.1.2_beta.rar" que está en el escritorio.



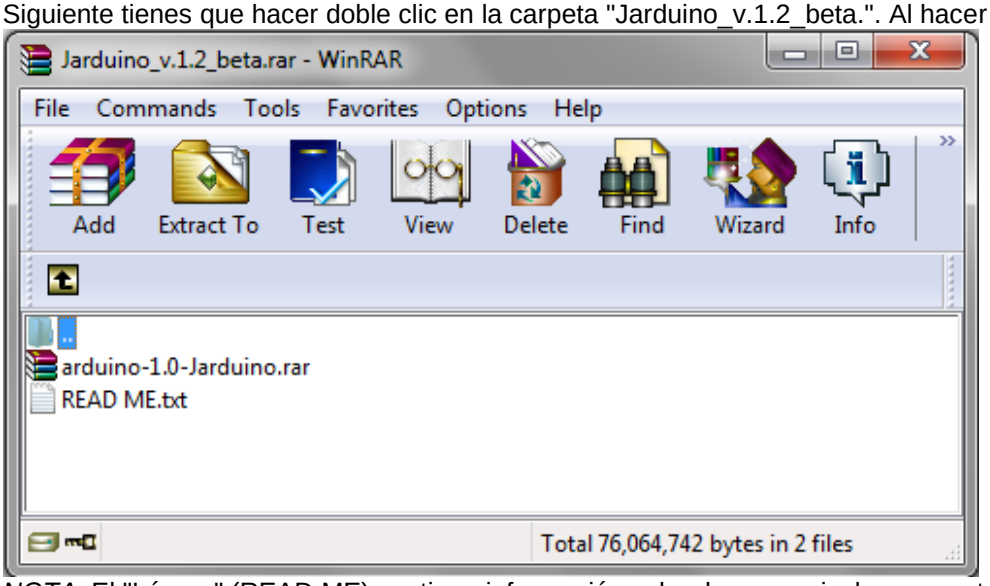
Haciendo esto nos lleva a la siguiente pantalla:



:

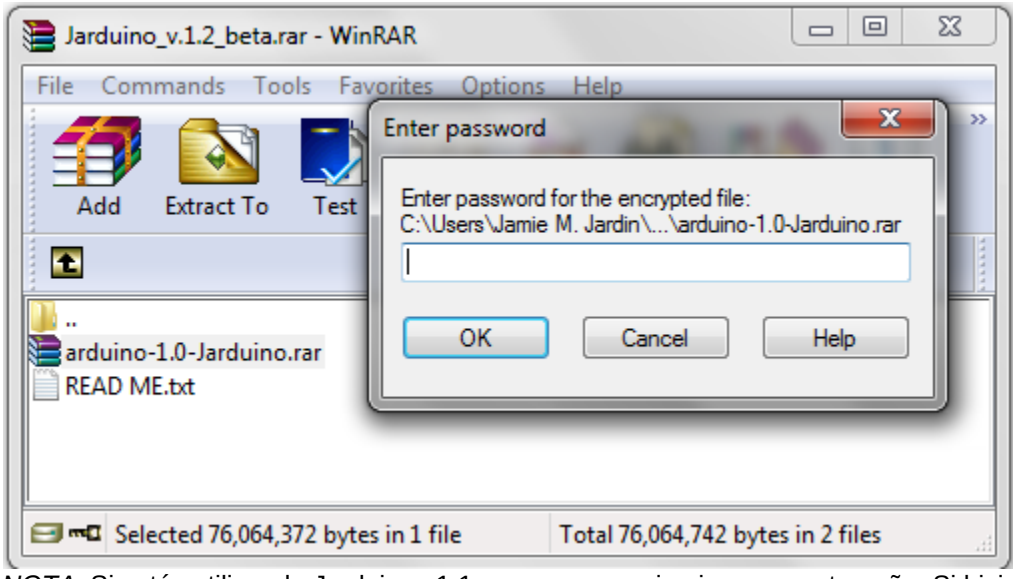
NOTA: El archivador que aparece en la foto de arriba es WinRAR, y por lo tanto puede ser ligeramente diferente dependiendo de tu archivador.

Siguiente tienes que hacer doble clic en la carpeta "Jarduino_v.1.2_beta.". Al hacer esto, verás esta pantalla:


Jarduino_v.1.2_beta.rar - WinRAR
File Commands Tools Favorites Options Help
Add Extract To Test View Delete Find Wizard Info
arduino-1.0-Jarduino.rar
READ ME.txt
Total 76,064,742 bytes in 2 files

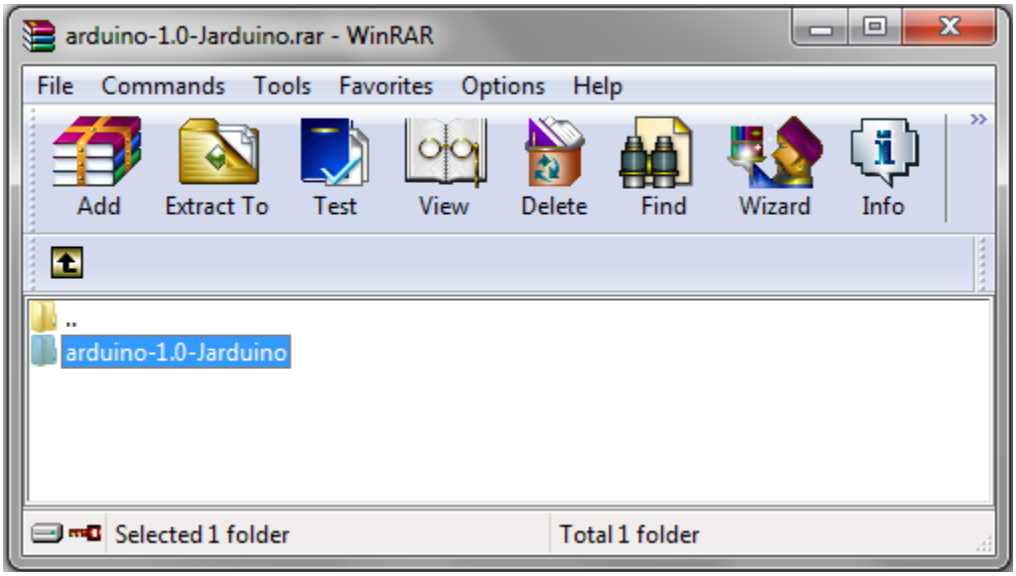
NOTA: El "Léame" (READ ME) contiene información sobre lo que se incluye en esta versión en particular.

Al hacer clic en el archivo "Arduino-1.0.Jarduino.rar" se abrirá un cuadro de diálogo que te pide que introduzcas una contraseña:

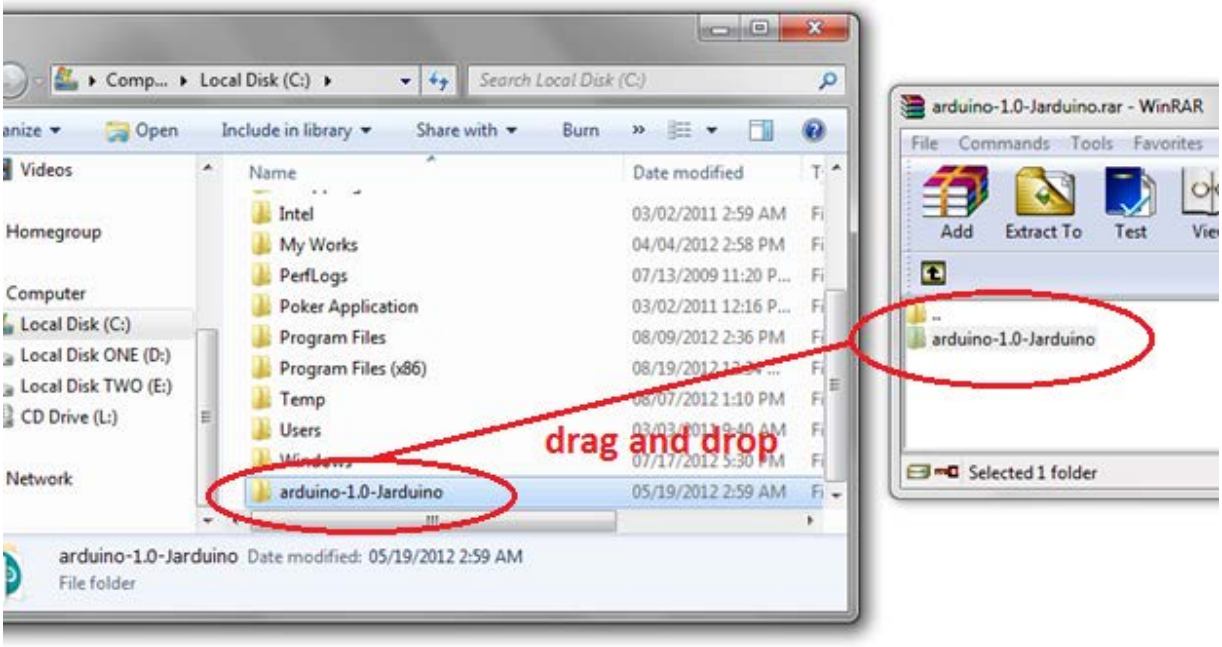


NOTA: Si estás utilizando Jarduino v1.1, no es necesaria ninguna contraseña. Si hiciste una donación razonable, te he puesto en mi lista de Beta Testers, y te envié por correo electrónico una contraseña para abrir el archivo.

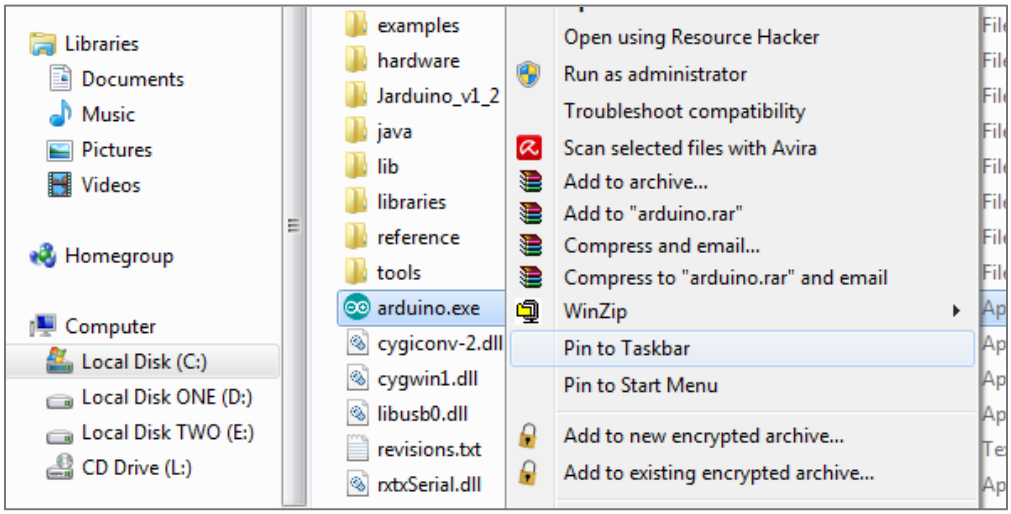
Después de introducir la contraseña (si es necesario), verás lo siguiente: El "Léame" contiene información sobre lo que se incluye en esta versión en particular.


arduino-1.0-Jarduino.rar - WinRAR
File Commands Tools Favorites Options Help
Add Extract To Test View Delete Find Wizard Info
arduino-1.0-Jarduino
Selected 1 folder
Total 1 folder

A continuación, todo lo que necesitas hacer es "arrastrar y soltar" esta carpeta a donde quieras que el Arduino quedé "instalado." Yo decidí ponerlo en mi disco C: /. Al hacer esto, se necesitará un poco de tiempo para que el archivador extraiga todos los archivos:



Es muy útil hacer un botón de "inicio rápido" para el IDE de Arduino. Para ello, "doble clic" en la carpeta recién creada "Arduino-1.0-Jarduino" para abrirla. A continuación, haga clic con el botón derecho en "arduino.exe" y haga clic en "Pin a la barra de tareas".



NOTA: Este procedimiento es para Windows 7, por lo que si estás usando alguna otra cosa, puede que tengas que tomar medidas diferentes para hacer un acceso directo.

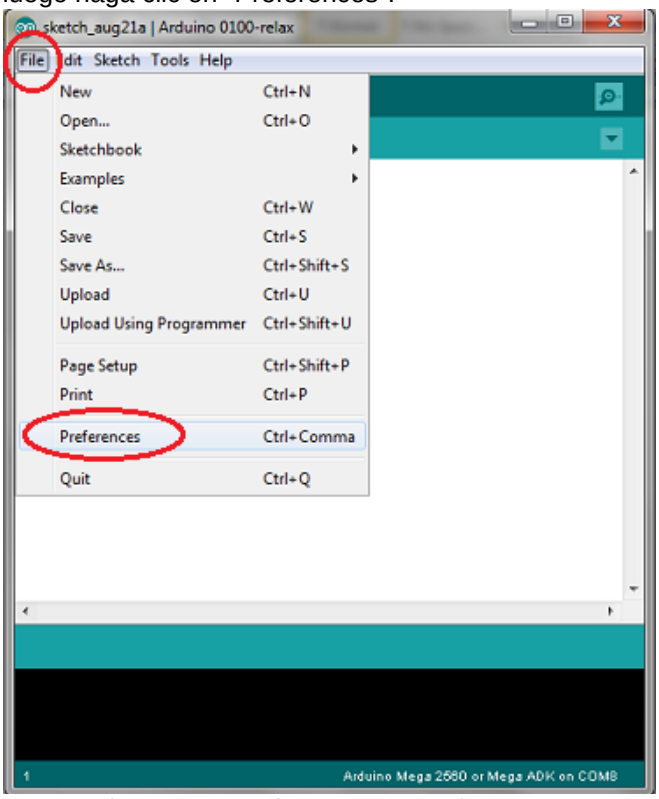
Ahora puedes fácilmente acceder al programa de Arduino desde la barra de tareas.

PASO 3: Conecta tu Arduino MEGA 2560

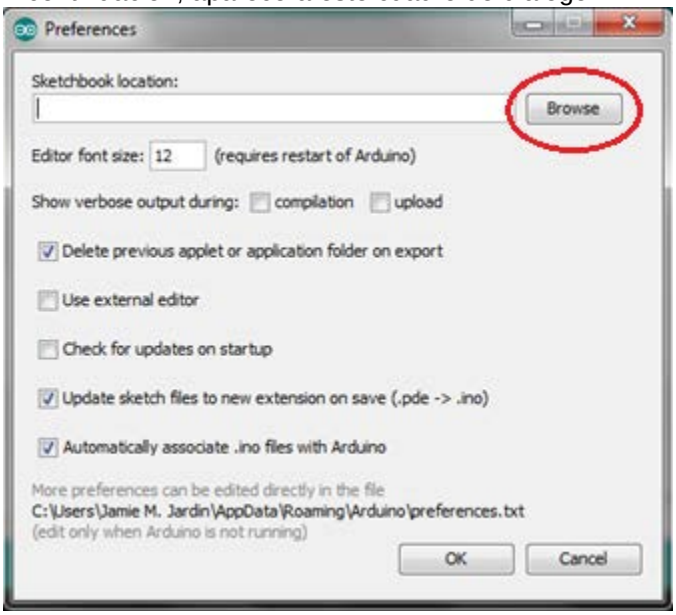
En este punto, conecta un extremo de un cable USB a un puerto USB en el ordenador y el otro extremo a la placa Arduino MEGA 2560. Si utilizas Windows (y está conectado a Internet), automáticamente debe descargar e instalar los controladores necesarios. Si estás teniendo problemas con este paso (es decir, - que tienes errores, los controladores no se han instalado correctamente, o si estás utilizando un sistema operativo diferente), por favor consulte el sitio web de Arduino para obtener sugerencias y soluciones de problemas de instalación).

PASO 4: Configurar el software de desarrollo Arduino

Primero, clic en el icono de la barra de tareas Arduino (si no lo has puesto a tu barra de tareas y haga clic en "arduino.exe" en la carpeta Arduino. El Arduino IDE debe cargar en pocos segundos. Cuando lo hace, haga clic en "File", luego haga clic en "Preferences".

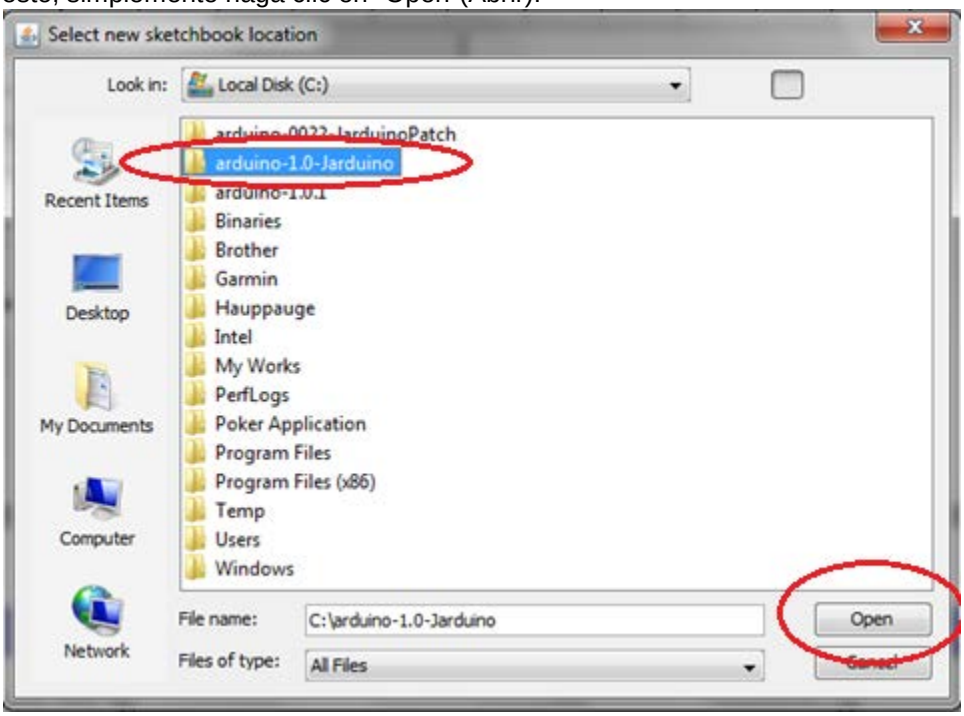


NOTA: Al cargar por primera vez, puede preguntar acerca de “sketchbook location”. No te preocupes por eso; Simplemente haz clic en "ok".
A continuación, aparecerá este cuadro de diálogo:

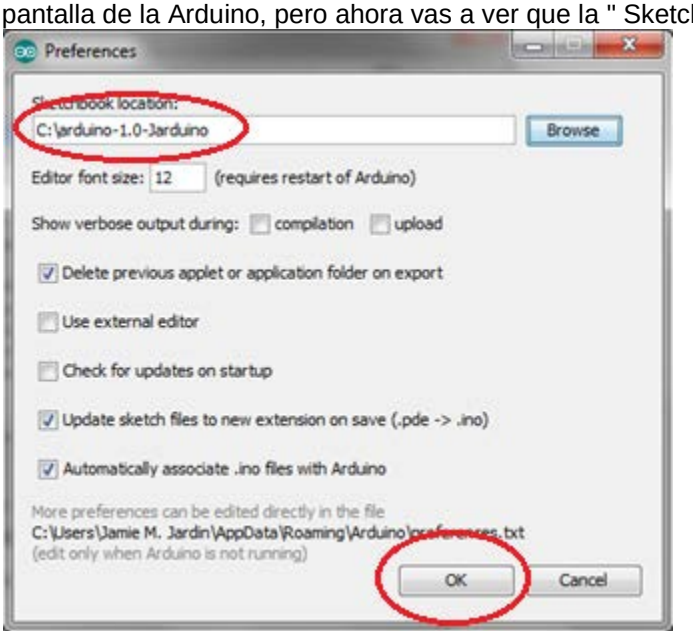

Preferences
Sketchbook location:
Browse
Editor font size: 12 (requires restart of Arduino)
Show verbose output during: compilation upload
Delete previous applet or application folder on export
Use external editor
Check for updates on startup
Update sketch files to new extension on save (.pde -> .ino)
Automatically associate .ino files with Arduino
More preferences can be edited directly in the file
C:\Users\Jamie M. Jardin\AppData\Roaming\Arduino\preferences.txt
(edit only when Arduino is not running)
OK
Cancel

NOTA: Y no te preocupes si la “sketchbook location” tiene algo lleno allí. Vas a cambiar esto rápidamente de todos modos...

Lo siguiente que hay que hacer es establecer la “sketchbook location”. Para ello, haga clic en “Browse” (Examinar) como he indicado anteriormente. El cuadro de explorador se debe abrir. Lo que quiero hacer es navegar a donde he guardado mi carpeta Arduino. Recordemos que yo tengo el mío puesto en la unidad C, así que voy a seguir adelante y navegar hasta allí. De un solo clic en la carpeta Arduino que quede destacado como es abajo. Una vez hecho esto, simplemente haga clic en "Open"(Abrir).

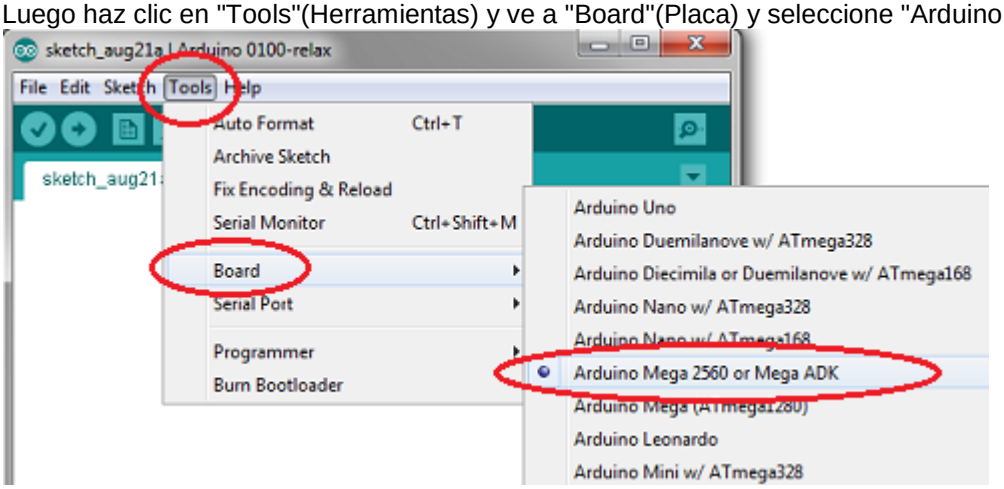


Al hacer clic en "Open"(Abrir), te llevará de nuevo a la opción "Preferences"(Preferencias) de la pantalla de la Arduino, pero ahora vas a ver que la " Sketchbook location" se ha cambiado a donde yo quiero que sea:

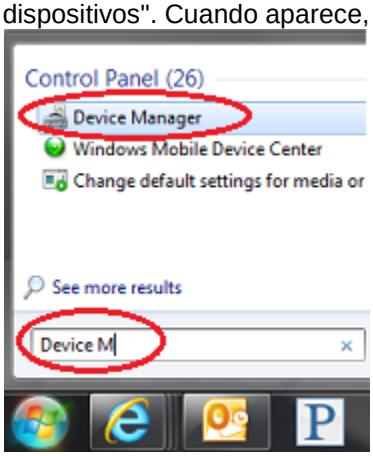


Simplemente haz clic en "OK” (Aceptar) y este cuadro de diálogo se cerrará.

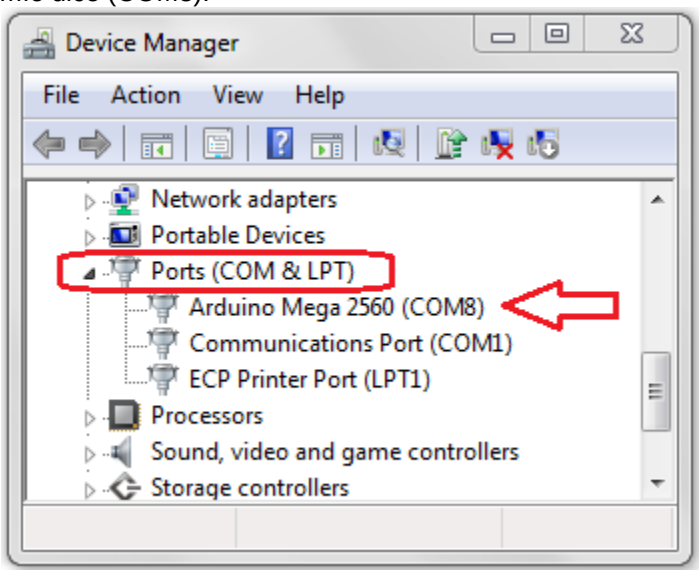
Luego haz clic en "Tools"(Herramientas) y ve a "Board"(Placa) y seleccione "Arduino MEGA 2560 o MEGA ADK".



Ahora tendrás que averiguar a qué puerto COM está enchufado tu Arduino. Si no lo has hecho ya, sigue adelante y conecta el Arduino al ordenador. A continuación, abra el "Administrador de dispositivos." En Windows 7, haga clic en el botón de inicio orbe azul, y en el comienzo de la barra de búsqueda escriba "Administrador de dispositivos". Cuando aparece, siga adelante y haga clic en él.

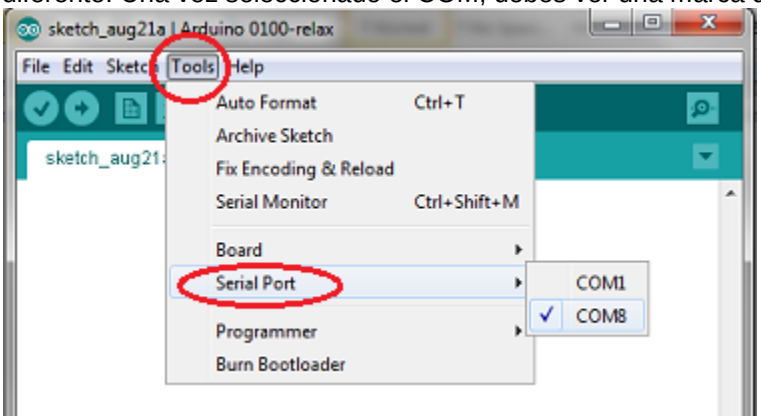


El Administrador de dispositivos se abrirá. Desplázate hacia abajo hasta donde dice "Puertos (COM & LPT) y haz click para expandirla. Debes ver a tu Arduino Mega 2560 en la lista. Junto a él se le ve entre paréntesis el número COM. El mío dice (COM8).



Puedes memorizar o escribir el número de puerto COM está utilizando el Arduino.

Luego de vuelta a la IDE Arduino, haz click en "Tools"(Herramientas) de nuevo, pero esta vez hay que navegar hacia abajo hasta donde dice "Serial Port". Aparecerá un cuadro que se deslizará hacia fuera donde puedes seleccionar el puerto COM donde está enchufado el Arduino. El mío está en "COM8." Es muy probable que el tuyo sea en un COM diferente. Una vez seleccionado el COM, debes ver una marca de verificación junto a él.



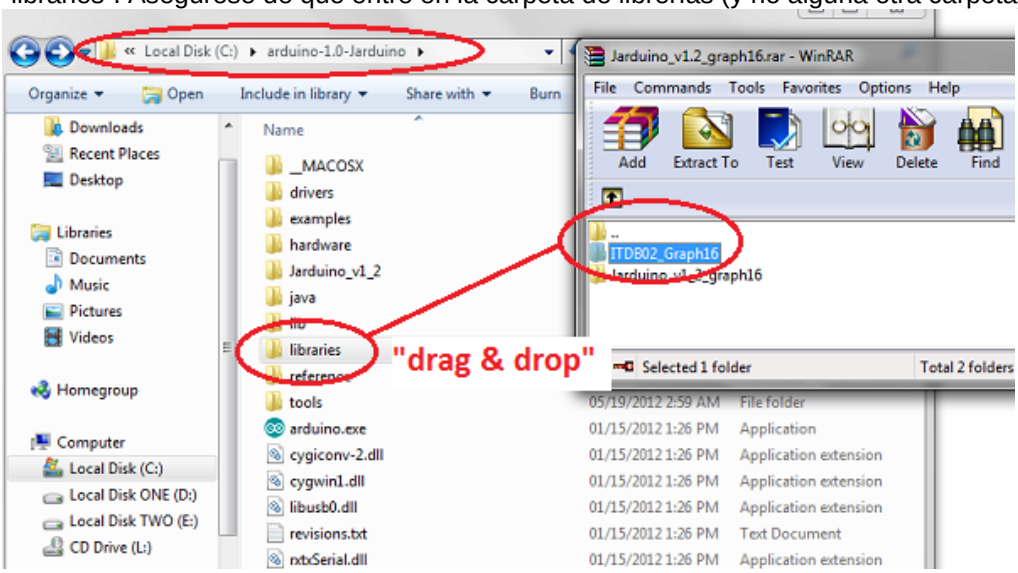
NOTA: Por último, y esto es muy importante: cerrar el IDE Arduino. Asegúrate de que no hay ventanas del Arduino que están abiertas.

PASO 5: Poner todos los archivos en los lugares adecuados

Arduino puede ser muy meticuloso cuando se trata de la ubicación de los archivos, y es probablemente la razón principal por la que se obtienen errores cuando la gente utiliza por primera vez el Arduino. Siempre y cuando los archivos están donde el IDE espera que sean, no debería haber demasiados problemas.

Ahora, si has estado siguiendo este tutorial palabra por palabra, entonces no deberías tener ningún problema con la ubicación de los archivos. Todo, desde esa descarga en particular (Jarduino v1.2 (beta1)) es una copia exacta de mi carpeta Arduino 1,0 (versión revisada) exactamente como estaba en mi equipo. Lo mismo es cierto para Jarduino v1.1 y el IDE Arduino 0022. Sin embargo, para subidas posteriores de los lanzamientos Jarduino, no incluí el IDE, y en algunos de ellos les adjunto una librería adicional aquí y allá. Por lo tanto, todos esos archivos tendrán que ser colocados en los lugares correctos para poder cargarse correctamente.

Para las diversas LIBRERIAS, tendrás que colocarlas en la carpeta “libraries” (librerías) del "arduino-1.0-Jarduino" (o cualquier versión del IDE que estés usando). Hay aquí un ejemplo de la colocación de la librería "ITDB02_Graph16" en el lugar adecuado. Comience por descargar la v1.2 Jarduino (beta1 con ITDB02_Graph16). Al llegar a la parte cuando se abre el archivo, simplemente "arrastrar y soltar", la carpeta "ITDB02_Graph16" en la carpeta "libraries". Asegúrese de que entre en la carpeta de librerías (y no alguna otra carpeta):



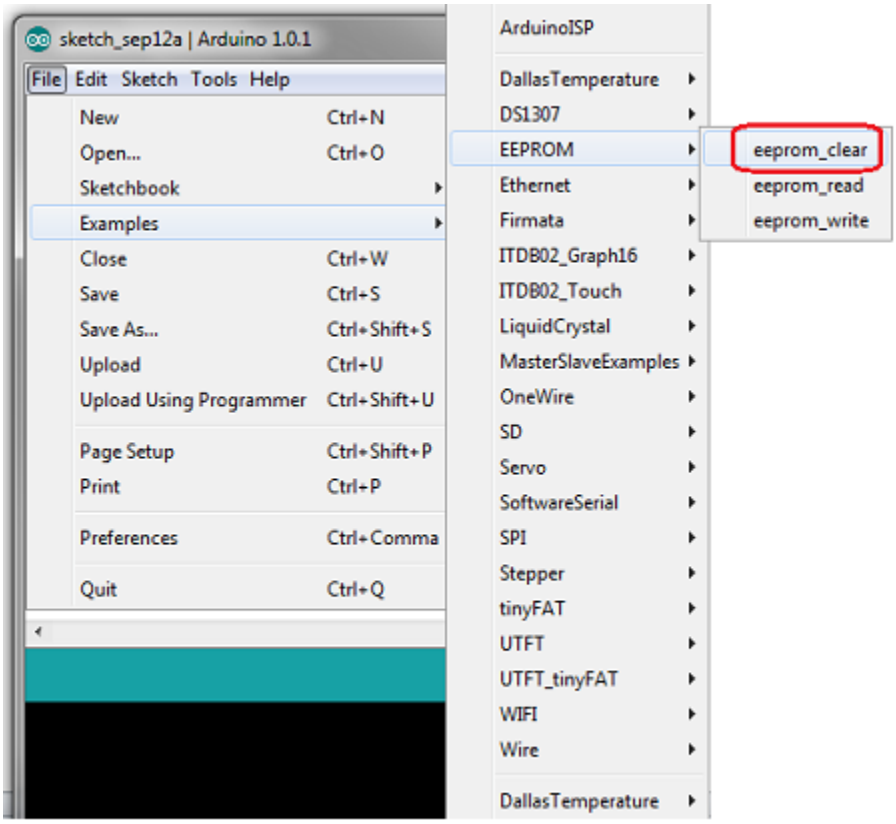
NOTA: Este es el mismo procedimiento que se deberá seguir para todas las bibliotecas subsiguientes que pueden añadirse en el futuro.

Para PROGRAM SKETCHES (es decir, - varias versiones del Jarduino), también se pueden colocar estos archivos en las "librerías" sub-carpetas. Yo sin embargo prefiero colocarlos directamente en la carpeta principal de la Arduino. Ya se puede ver en la foto de arriba que es la ubicación de "Jarduino_v1_2." Aquí es donde me gustaría extraer "Jarduino_v1_2_graph16" si quería subir ese dibujo especial en mi placa Arduino.
OTRA NOTA MUY IMPORTANTE: Cada vez que se realiza un cambio en la carpeta principal del Arduino, como la colocación de un nuevo sketch, o poner en una carpeta una nueva biblioteca, asegúrese de que el IDE de Arduino está cerrado! A pesar de que se pueden hacer cambios, mientras que el IDE está abierto, los cambios que realice no serán reconocidos hasta que la librería ha sido reconstruida. Para que eso ocurra, el IDE de Arduino tiene que "construirla". Lo hace cuando el IDE se carga (o en el caso de ser abierto, al cerrar y volver a abrir el IDE, se "re-construirá" la librería).

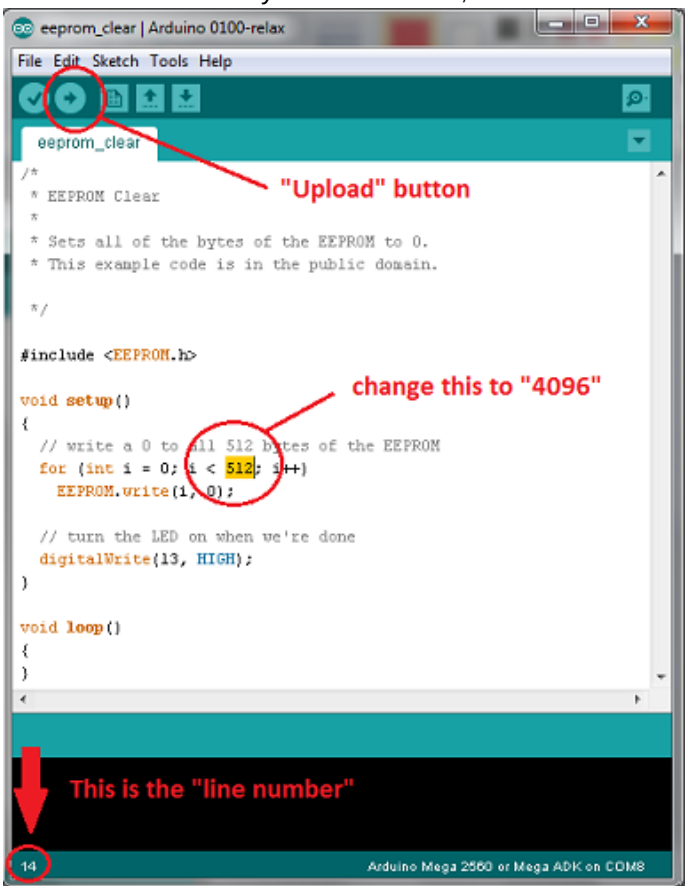
PASO 6: BORRAR LA MEMORIA EEPROM

Ahora lo que hay que hacer es asegurarse de que la EEPROM (alias la Memoria de Arduino) se borra. Tendrá que borrar la memoria cada vez que se decide cambiar las versiones de la Jarduino. Si simplemente quieres volver a cargar la misma versión, no es necesario que realice este paso (a pesar de que está perfectamente bien si lo haces).

En primer lugar, abrir el IDE de Arduino. Haga clic en “File” ("Archivo"), baje hasta “Examples” ("Ejemplos"), después de más a "EEPROM", y finalmente haga clic en “eeprom_clear ".

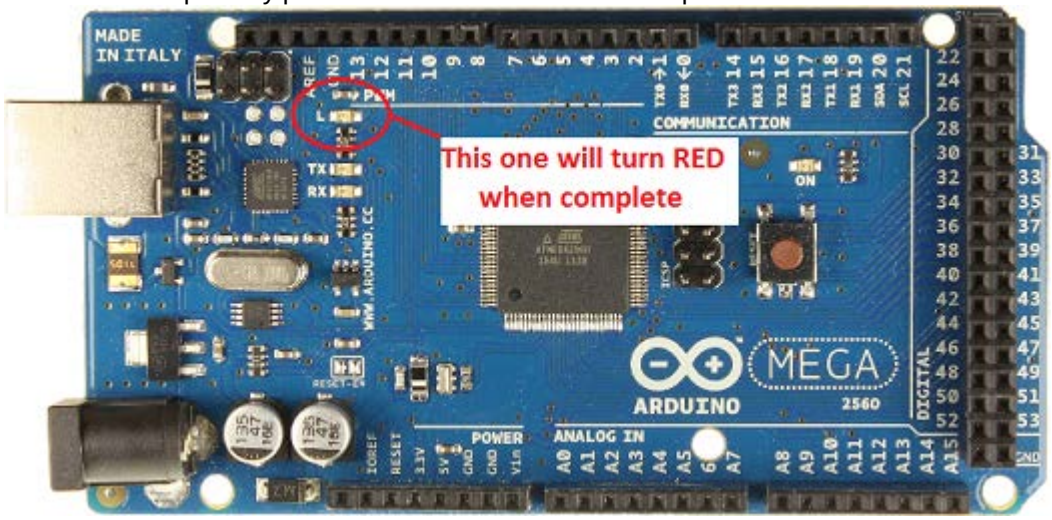


Al hacer clic en "eeprom_clear" se abrirá el ejemplo "eeprom_clear". Por defecto, este esquema tiene un valor guardado de "512". Esto es para el Arduino UNO / Duemilanove que sólo tiene 512 bytes de memoria. El Arduino Mega 2560 por otro lado tiene 4096 bytes. Por lo tanto, es necesario cambiar eso de "512" a "4096".



SUGERENCIA: al hacer clic en cualquier lugar dentro de un esquema permitirá conocer el número de línea en la parte inferior izquierda del IDE Arduino

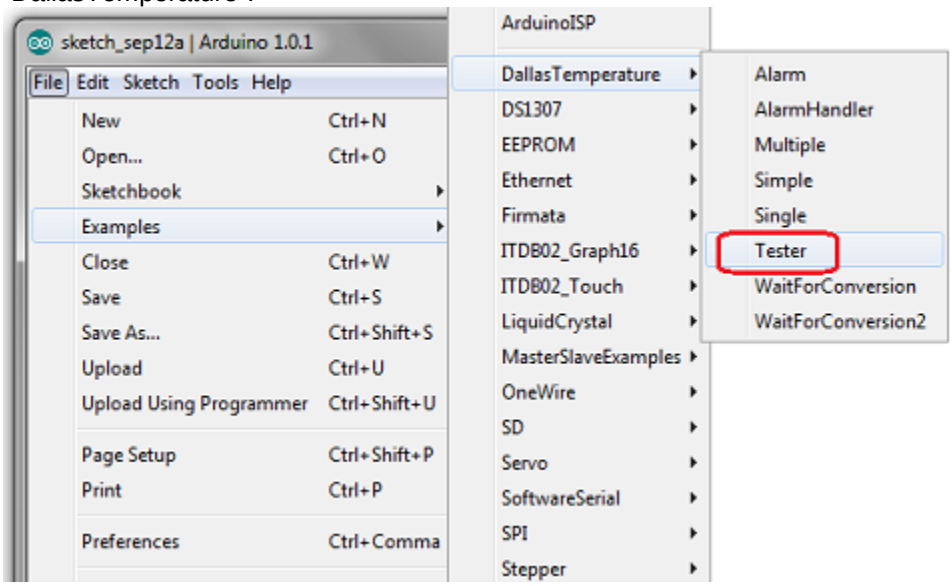
Después de cambiar el "512" a "4096", proceda a hacer clic en el botón "Upload"(""Cargar"), que se indica más arriba. El "clear_eeprom" sólo tarda unos pocos segundos para cargar al Arduino. Mientras que se carga, el "RX", "TX" y "L" LED parpadearán rápidamente. Una vez completada la carga, el LED dejará de parpadear y el programa se ejecutará hasta que se haya "limpiado" toda la memoria. Sabrá que ha terminado el trabajo cuando la Red "L" se enciende en la placa y permanecerá encendido. Todo el proceso sólo debe tardar unos 15 segundos más o menos:



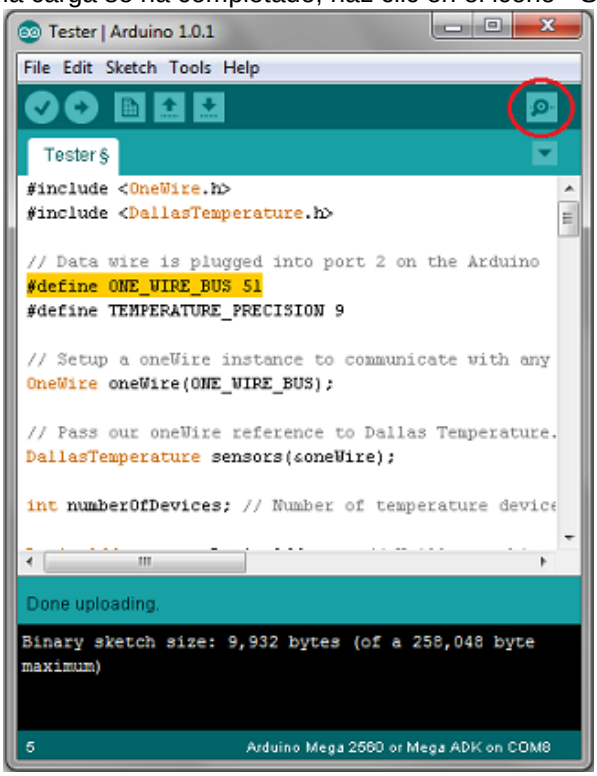
PASO 7: Obtener los números de serie de los sensores de temperatura

El Jarduino utiliza tres sensores de temperatura. En particular, se utilizan los Dallas DS18B20 termómetros digitales de temperatura. Cada uno de estos sensores tiene su número de serie (adress) propio y único. Debido a esto, puedes poner tantos sensores como deseas en un solo cable. El Jarduino simplemente llama a la serie de cualquier sensor que quiere información de temperatura, y el sensor dará al Jarduino la información. Consulte esta sección para ver como conectar los sensores al Arduino.

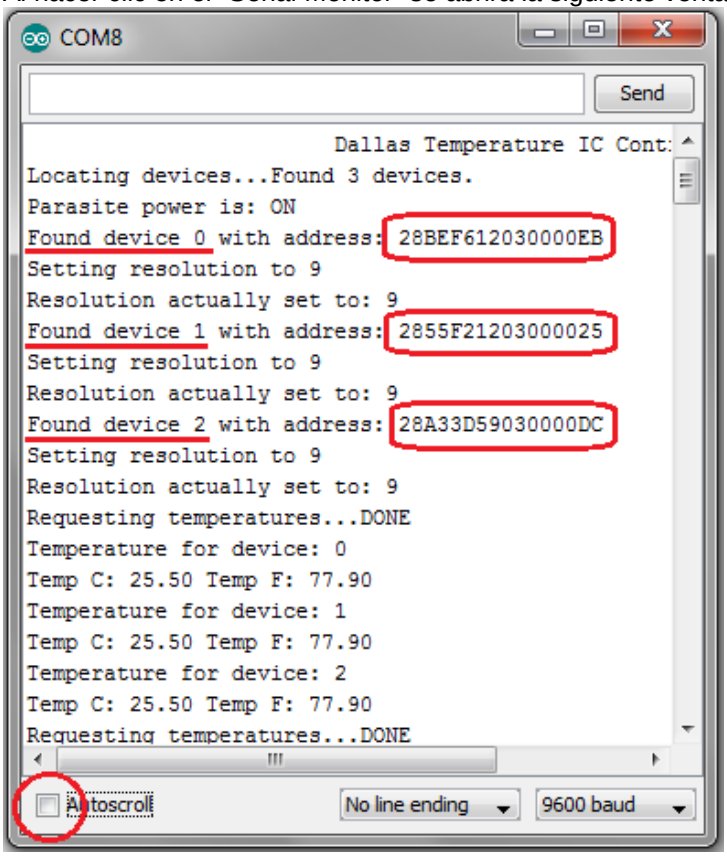
OK, con los sensores correctamente conectado a la placa Arduino, abrir el ejemplo "Tester", ubicado en el la librería "DallasTemperature":



Cuando el ejemplo se abre, cambie la línea 5 "# define ONE_WIRE_BUS 2" a "# define ONE_WIRE_BUS 51" y sigue adelante para cargarlo a tu Arduino (de la misma manera que has cargado el ejemplo "eeprom_clear". Cuando la carga se ha completado, haz clic en el icono "Serial Monitor" icono de la lupa, como se indica a continuación:


Tester | Arduino 1.0.1
File Edit Sketch Tools Help
Tester §
#include <OneWire.h>
#include <DallasTemperature.h>
// Data wire is plugged into port 2 on the Arduino
#define ONE_WIRE_BUS 51
#define TEMPERATURE_PRECISION 9
// Setup a oneWire instance to communicate with any
OneWire oneWire(ONE_WIRE_BUS);
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&oneWire);
int numberOfDevices; // Number of temperature device
Done uploading.
Binary sketch size: 9,932 bytes (of a 258,048 byte maximum)
5
Arduino Mega 2560 or Mega ADK on COM8

Al hacer clic en el "Serial Monitor" se abrirá la siguiente ventana:


COM8
Send
Dallas Temperature IC Cont:
Locating devices...Found 3 devices.
Parasite power is: ON
Found device 0 with address: 28BEF612030000EB
Setting resolution to 9
Resolution actually set to: 9
Found device 1 with address: 2855F21203000025
Setting resolution to 9
Resolution actually set to: 9
Found device 2 with address: 28A33D59030000DC
Setting resolution to 9
Resolution actually set to: 9
Requesting temperatures...DONE
Temperature for device: 0
Temp C: 25.50 Temp F: 77.90
Temperature for device: 1
Temp C: 25.50 Temp F: 77.90
Temperature for device: 2
Temp C: 25.50 Temp F: 77.90
Requesting temperatures...DONE
Autoscroll
No line ending
9600 baud

La parte inferior izquierda tiene el “Autoscroll” (“Auto desplazamiento") marcado por defecto. Por el momento, seguir adelante y desmarcar. Asegúrate de ir a la parte superior de la ventana. Se puede ver que encontré a los tres sensores de temperatura, y el número de serie de cada uno. Puedes "copiar" y "pegar" estos números de serie en un archivo de texto y guardarlo para más tarde, ya que los necesitarás para el código del Jarduino.

Por lo tanto, estas son los tres números de serie que encontré por los míos:

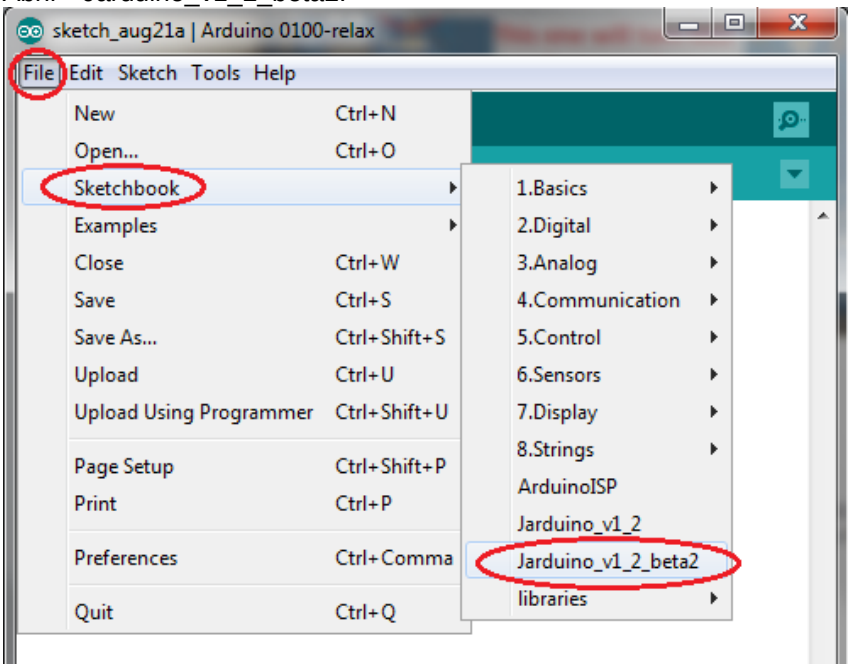
- 28BEF612030000EB
- 2855F21203000025
- 28A33D59030000DC

SUGERENCIA: Como tienes varios sensores conectados al mismo tiempo, puedes pensar que podría ser un poco difícil determinar cuál es el dispositivo al que pertenece cada número de serie Siempre se puede hacer uno a la vez. Si lo prefieres, puede volver a marcar la casilla "Desplazamiento automático". Luego, apriete una de la sonda de temperatura entre los dedos. Espere unos segundos y, a continuación, mientras mantiene el sensor, quite la marca de la "Auto-Desplazamiento." Uno de los sensores debe ser significativamente diferente del resto. Anote el número del sensor, desplácese de nuevo a la cima, y ya sabes la dirección para ese sensor. Etiquetar el sensor con un trozo de cinta adhesiva, y luego déjelo a un lado. Repita este procedimiento para los otros sensores.

PASO 8: Cargar el código del Jarduino

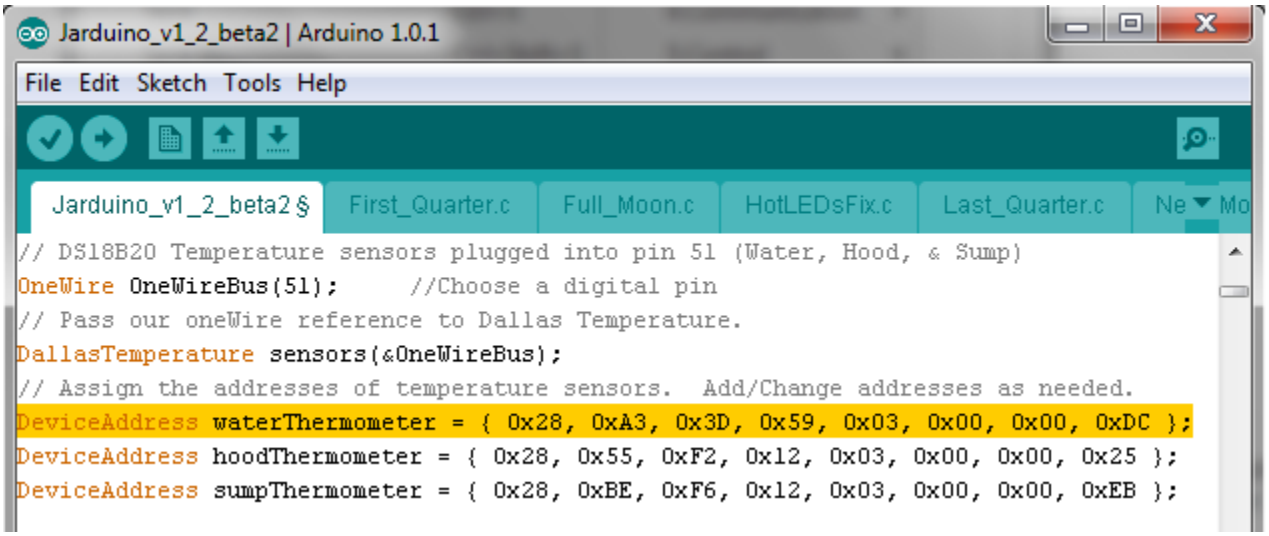
Ahora que la memoria ha sido borrada, y tienes las direcciones de tus sensores de temperatura, la placa Arduino está lista para aceptar el código del Jarduino. Para este ejemplo, voy a cargar Jarduino v1.2 (beta 2). En este punto, ya debe saber dónde colocar todos los archivos.

Abrir "Jarduino_v1_2_beta2."



NOTA: Esta versión particular utiliza la librería "ITDB02_Graph16". Puedes "fácilmente" hacer que esto funcione con la "librería UTFT" si es necesario. Simplemente hay que cambiar las líneas 131 y 147 de este código para parecerse a los mismos números de línea en "Jarduino_v1_2".

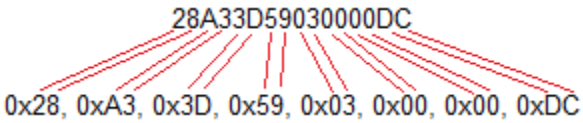
Una vez que tenga el código Jarduino abierto, tendrás que hacer algunos cambios. Ya que has descubierto los números de serie de tus sensores, adelante e introduzca esos valores en el código. Desplázate hacia abajo a las líneas 216-218:


Jarduino_v1_2_beta2 | Arduino 1.0.1
File Edit Sketch Tools Help
Jarduino_v1_2_beta2 §
First_Quarter.c
Full_Moon.c
HotLEDsFix.c
Last_Quarter.c
// DS18B20 Temperature sensors plugged into pin 51 (Water, Hood, & Sump)
OneWire OneWireBus(51); //Choose a digital pin
// Pass our oneWire reference to Dallas Temperature.
DallasTemperature sensors(&OneWireBus);
// Assign the addresses of temperature sensors. Add/Change addresses as needed.
DeviceAddress waterThermometer = { 0x28, 0xA3, 0x3D, 0x59, 0x03, 0x00, 0x00, 0xDC };
DeviceAddress hoodThermometer = { 0x28, 0x55, 0xF2, 0x12, 0x03, 0x00, 0x00, 0x25 };
DeviceAddress sumpThermometer = { 0x28, 0xBE, 0xF6, 0x12, 0x03, 0x00, 0x00, 0xEB };

A primera vista, estos no se parecen a los números de serie que tenemos, pero vamos a echar un vistazo más de cerca. Recordemos que los números de serie que me dieron para mis sensores han sido:

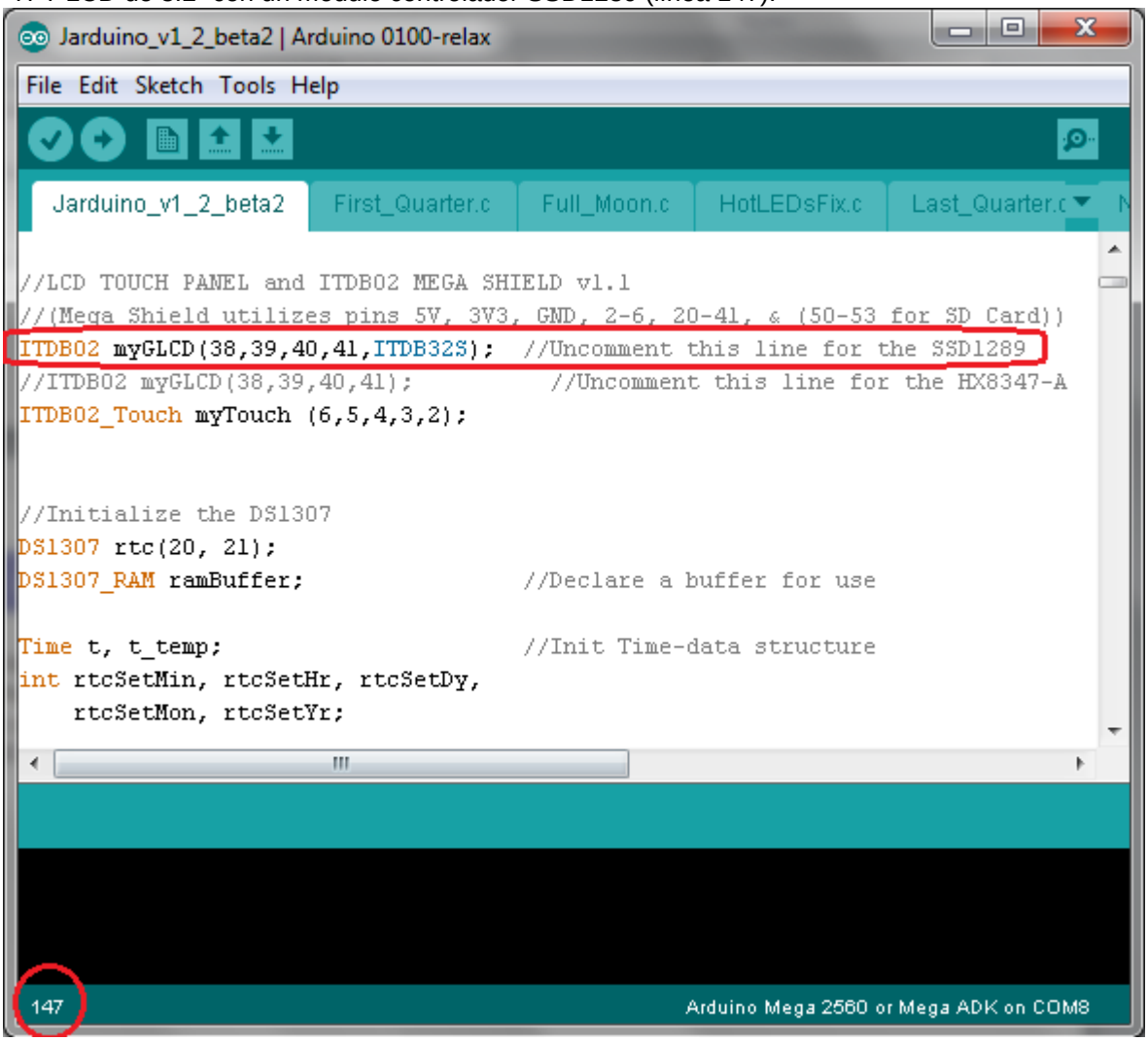
- 28BEF612030000EB
- 2855F21203000025
- 28A33D59030000DC

Si nos centramos en el último número de serie (que yo designo como mi termómetro de agua), lo voy a cambiar a un formato que el IDE puede entender. Hago esto agrupándolos en parejas de dos y añadir un "0x" delante de cada par, separados por una coma:

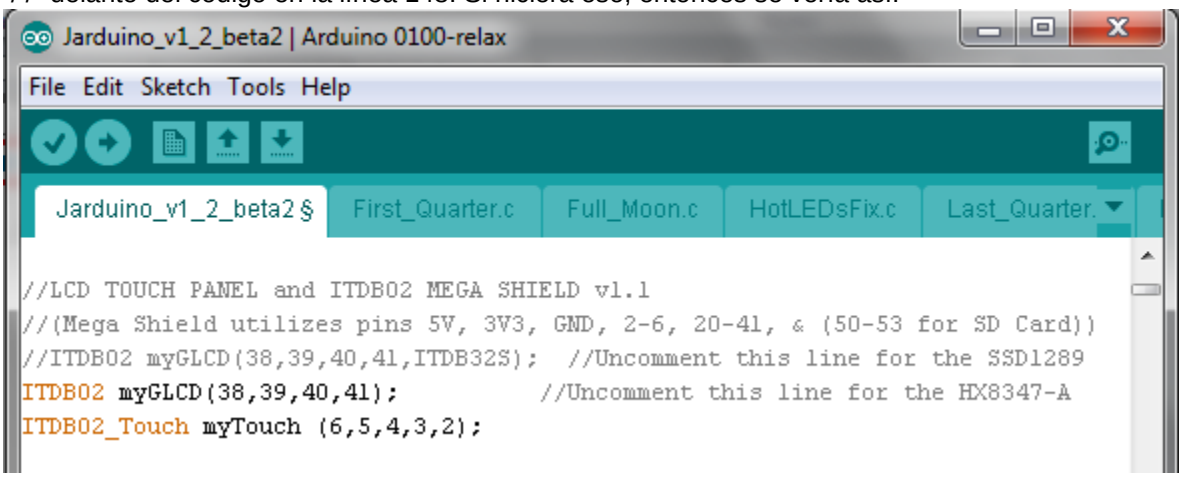


Ahora todo lo que tienes que hacer es poner los números de serie obtenidos del DallasTemperature "Tester" sketch en el formato correcto, y copiar los de más al Jarduino.

Ahora que ya tiene los números de serie de los sensores de temperatura, tendrás que asegurarte de que tienes el módulo LCD seleccionado correctamente. Para este ejemplo, voy a mostrar cómo el código debe leer una pantalla "TFT LCD de 3.2" con un módulo controlador SSD1289 (línea 147).

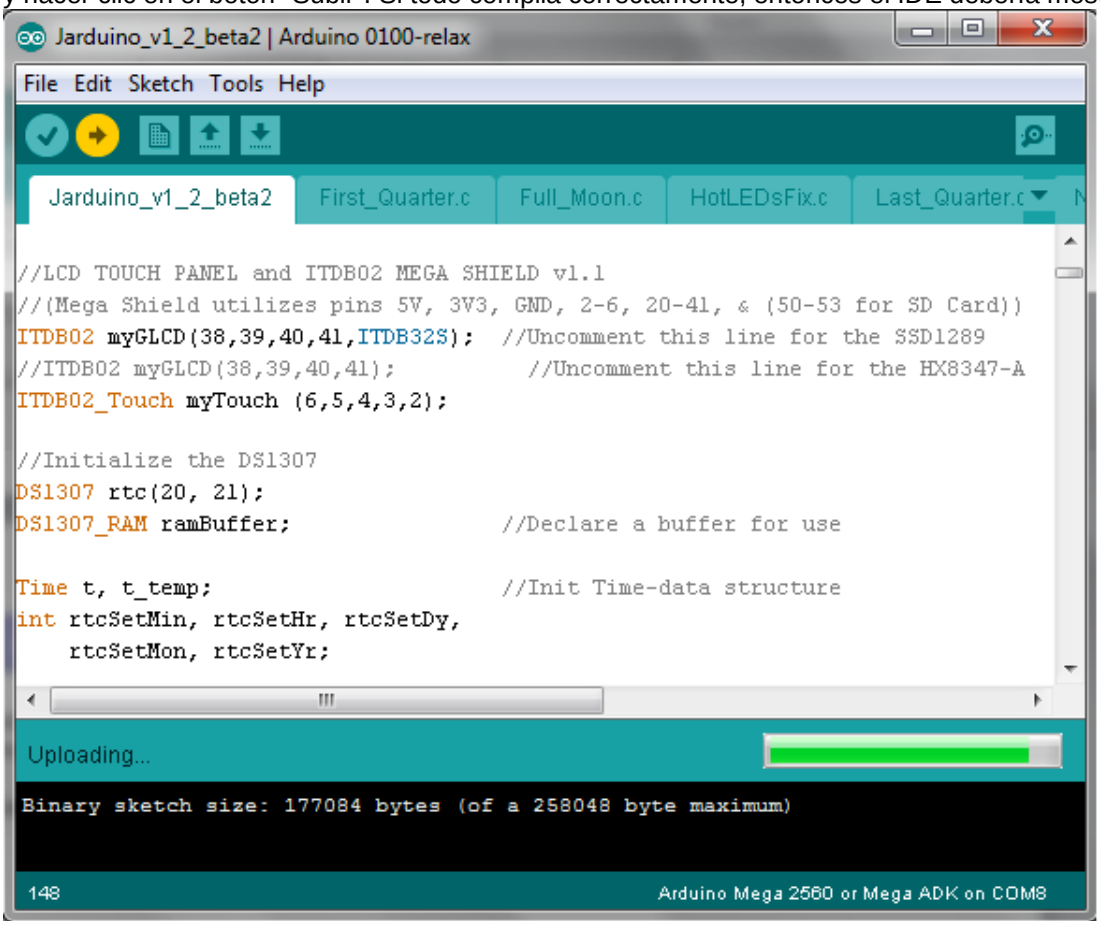

Jarduino_v1_2_beta2 | Arduino 0100-relax
File Edit Sketch Tools Help
Jarduino_v1_2_beta2
First_Quarter.c
Full_Moon.c
HotLEDsFix.c
Last_Quarter.c
//LCD TOUCH PANEL and ITDB02 MEGA SHIELD v1.1
//(Mega Shield utilizes pins 5V, 3V3, GND, 2-6, 20-41, & (50-53 for SD Card))
ITDB02 myGLCD(38,39,40,41,ITDB32S); //Uncomment this line for the SSD1289
//ITDB02 myGLCD(38,39,40,41); //Uncomment this line for the HX8347-A
ITDB02_Touch myTouch (6,5,4,3,2);
//Initialize the DS1307
DS1307 rtc(20, 21);
DS1307_RAM ramBuffer; //Declare a buffer for use
Time t, t_temp; //Init Time-data structure
int rtcSetMin, rtcSetHr, rtcSetDy,
rtcSetMon, rtcSetYr;
147
Arduino Mega 2560 or Mega ADK on COM8

Como se puede ver, en realidad sí tenemos seleccionado el correcto. La otra opción en la línea 148 es para el módulo "HX8347-A". Se puede ver qué línea de código en particular es "gris" hacia fuera. En el IDE de Arduino, poniendo un "/ /" delante de una línea de código, hace que esta línea sea "invisible" para el Arduino. Si se me ocurrió utilizar una "HX8347-A," simplemente ponemos "/ /" delante del código en la línea 147 (que va a "gris" hacia fuera), y me gustaría borrar los "/ /" delante del código en la línea 148. Si hiciera eso, entonces se vería así:


Jarduino_v1_2_beta2 | Arduino 0100-relax
File Edit Sketch Tools Help
Jarduino_v1_2_beta2 §
First_Quarter.c
Full_Moon.c
HotLEDsFix.c
Last_Quarter.
//LCD TOUCH PANEL and ITDB02 MEGA SHIELD v1.1
//(Mega Shield utilizes pins 5V, 3V3, GND, 2-6, 20-41, & (50-53 for SD Card))
//ITDB02 myGLCD(38,39,40,41,ITDB32S); //Uncomment this line for the SSD1289
ITDB02 myGLCD(38,39,40,41); //Uncomment this line for the HX8347-A
ITDB02_Touch myTouch (6,5,4,3,2);

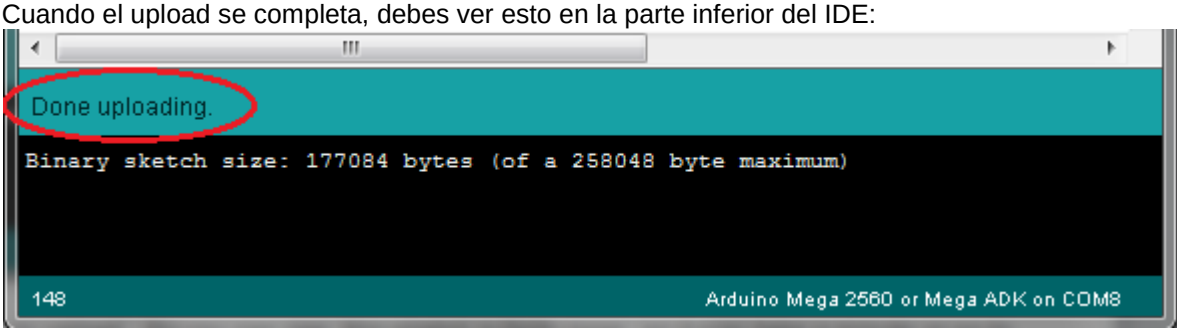
La librería ITDB02_graph16 tiene el controlador HX8347-A predeterminado, y por lo tanto no requiere que especifiques el número de modelo (es decir, - ITDB32S para la SSD1289). Para un controlador que no sea uno de esos dos, es posible que tenga que consultar la Tabla de compatibilidad en la sección de hardware para obtener el código necesario. El código que puede ser necesario aparece bajo el título “Model for ITDB02() ” ("Modelo para ITDB02 ()."). La selección del modelo equivocado hará que el Jarduino mostrara una pantalla blanca.

Volviendo al ejemplo, ahora que ya hemos seleccionado el módulo correcto que vamos a usar, vamos a seguir adelante y hacer clic en el botón "Subir". Si todo compila correctamente, entonces el IDE debería mostrar algo como esto:


Jarduino_v1_2_beta2 | Arduino 0100-relax
File Edit Sketch Tools Help
Jarduino_v1_2_beta2
First_Quarter.c
Full_Moon.c
HotLEDsFix.c
Last_Quarter.c
//LCD TOUCH PANEL and ITDB02 MEGA SHIELD v1.1
//(Mega Shield utilizes pins 5V, 3V3, GND, 2-6, 20-41, & (50-53 for SD Card))
ITDB02 myGLCD(38,39,40,41,ITDB32S); //Uncomment this line for the SSD1289
//ITDB02 myGLCD(38,39,40,41); //Uncomment this line for the HX8347-A
ITDB02_Touch myTouch (6,5,4,3,2);
//Initialize the DS1307
DS1307 rtc(20, 21);
DS1307_RAM ramBuffer; //Declare a buffer for use
Time t, t_temp; //Init Time-data structure
int rtcSetMin, rtcSetHr, rtcSetDy,
rtcSetMon, rtcSetYr;
Uploading...
Binary sketch size: 177084 bytes (of a 258048 byte maximum)
148
Arduino Mega 2560 or Mega ADK on COM8

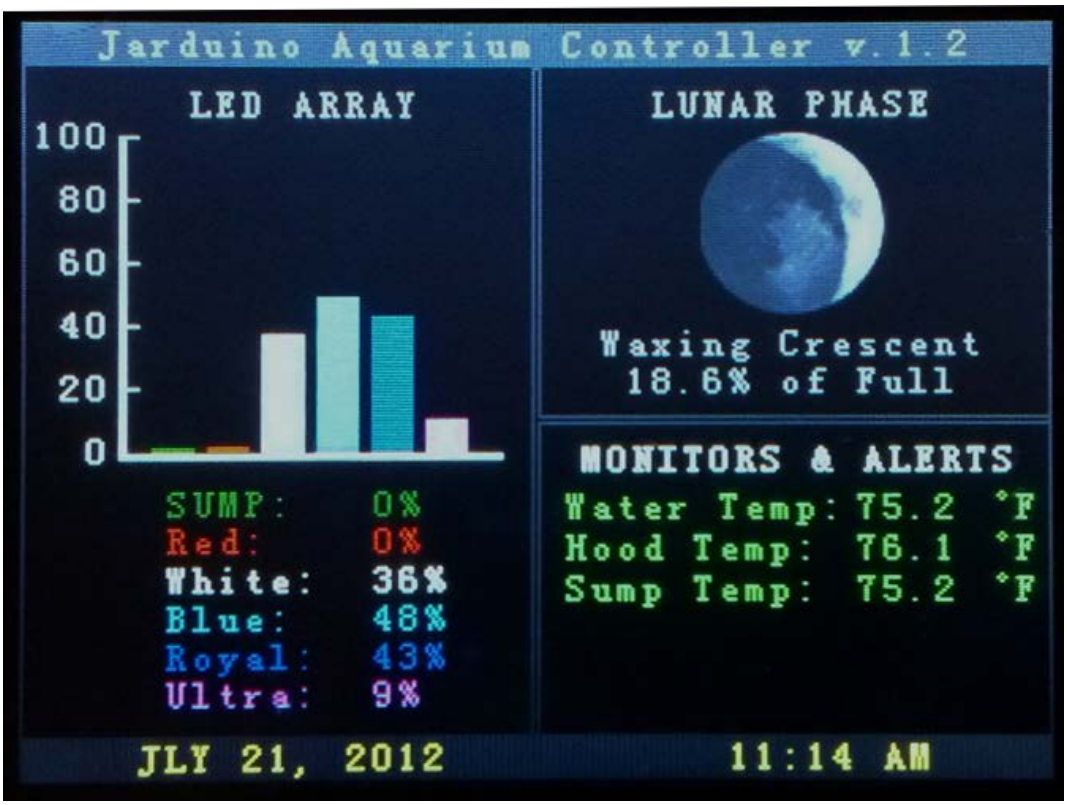
NOTA: No te preocupes si tarda un poco en cargar. Como se puede ver, este sketch es bastante grande, por lo que puede tardar un minuto más o menos para subir. Si va mucho más tiempo que eso, entonces puedes tener algunos problemas que debes atender. Lo más probable los uploads que se quedan "atrapados"suelen ser culpa del gestor de arranque o el IDE. Por favor, póngase en contacto conmigo si esto sucede, y voy a tratar de ayudarte. Vale la pena mencionar que este problema puede surgir si se usa Arduino 1.0.1, así que tome nota.

Cuando el upload se completa, debes ver esto en la parte inferior del IDE:



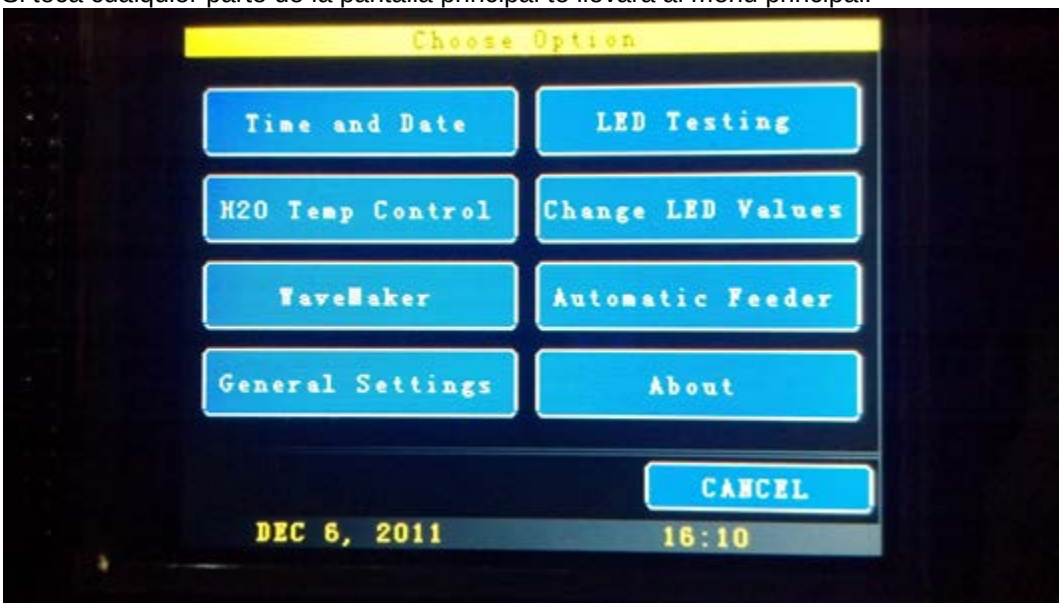
Con suerte, todo salió bien y se ve la pantalla principal de la Jarduino en la pantalla LCD (como aparece en la portada de este manual).

4. JARDUINO FUNCTIONALIDAD



Como se puede ver, la pantalla principal muestra la salida de cada color LED y el porcentaje actual. También se incluye la Fase Lunar completa con una foto de la fase actual, una descripción de dicha fase, y el porcentaje actual de una luna llena. Bajo la sección MONITORS & ALERTS (Monitores Y Alertas), la temperatura del agua y las temperaturas de la pantalla principal y la pantalla del sumidero son monitorizadas. Las alertas también parpadearán y cambiarán de color, etc, si / cuando las temperaturas salen fuera de los rangos definidos por el usuario. Si el sensor de temperatura no está conectado (o hay una mala conexión), aparecerá un mensaje de Error que se muestra en lugar de la temperatura. También puede encontrar este breve mensaje al regresar a esta pantalla. La barra de estado en la parte inferior de la pantalla donde se muestra la fecha y la hora aparece en todas las otras páginas y es completamente funcional en cada página subsiguiente.

Si toca cualquier parte de la pantalla principal te llevará al menú principal:

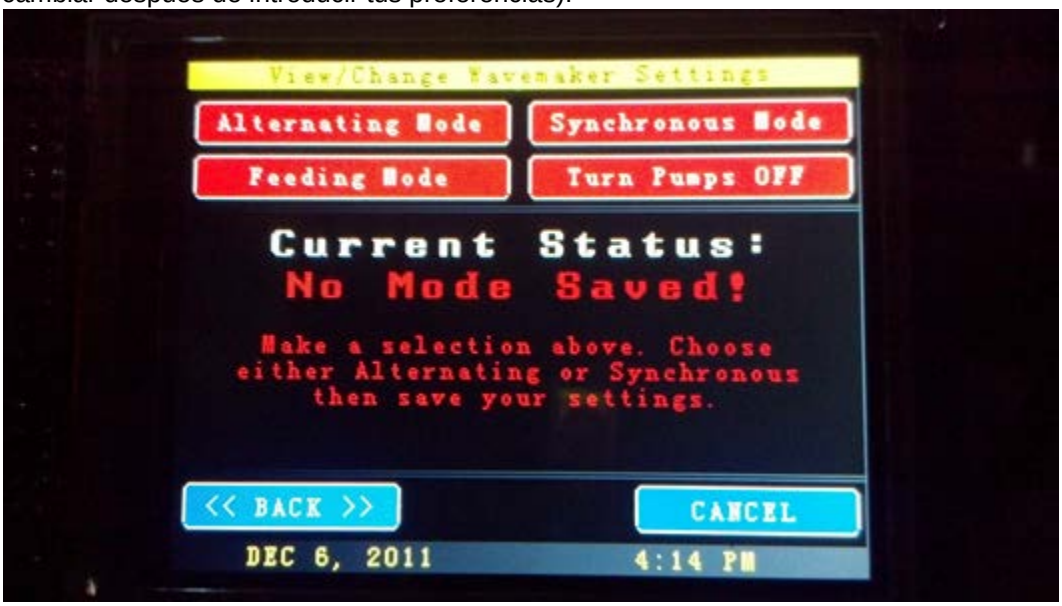


La opción **Time and Date** abrirá la siguiente pantalla:

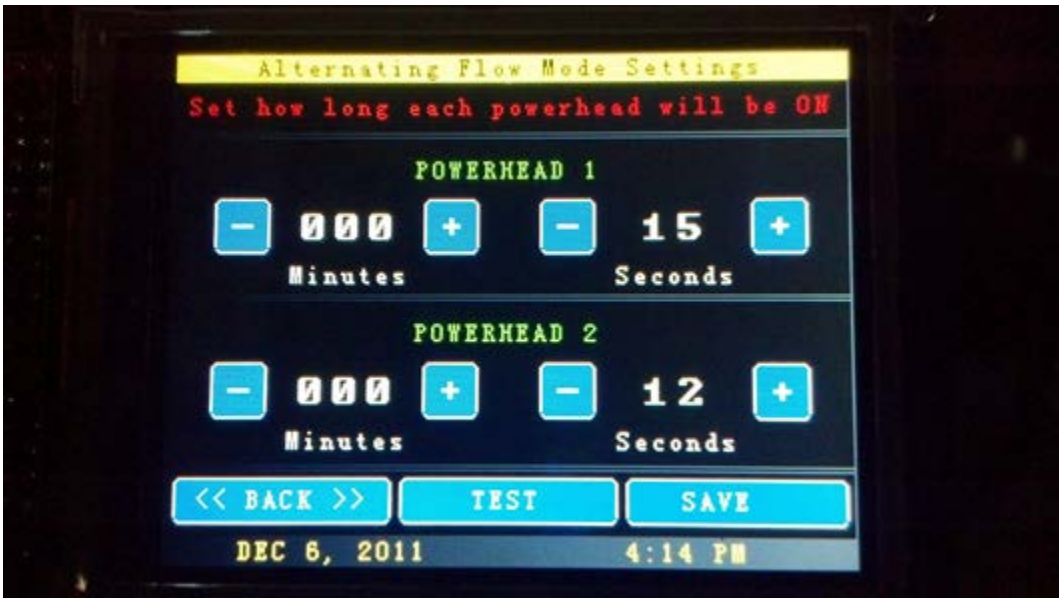
La opción **H2O Temp Control** se utiliza para configurar la temperatura deseada del agua, y también le permite establecer compensaciones para encender y apagar los calentadores y / o enfriadores, y también una alarma de audio:

Esto simplemente dice que si la temperatura del agua sube por encima los 79 grados F más de 2 grados F, se activara un enfriador. Si está por debajo de 2 grados F un activara un calentador. + / - 3,5 grados y una alarma sonará. Esta alarma también parpadeará en la pantalla principal y cambiara la lectura de la temperatura del agua de color verde a azul si el agua es demasiado frío o de verde a rojo si el agua es demasiado caliente.

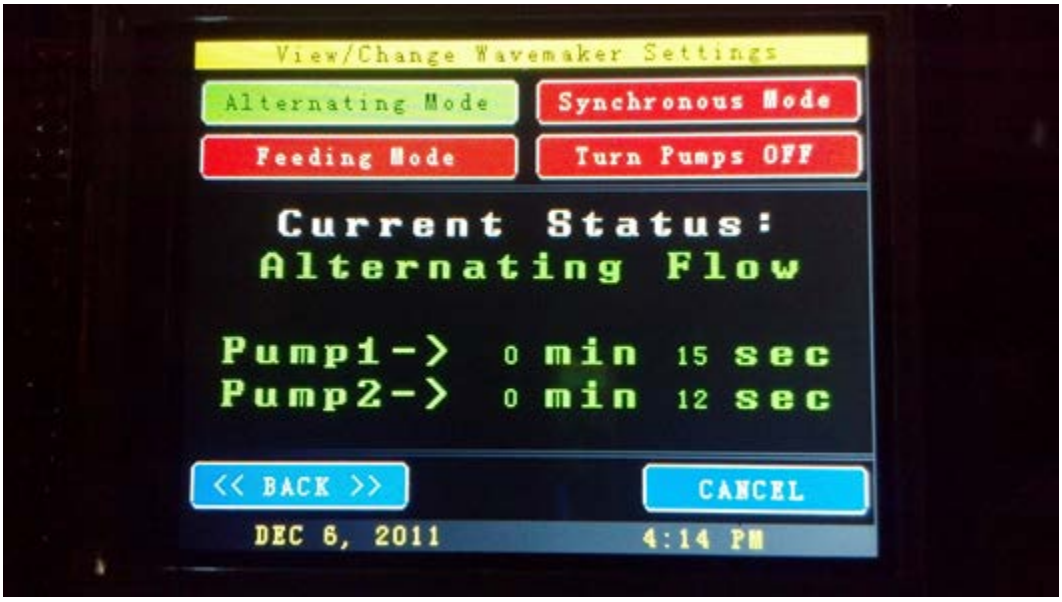
La siguiente es la pantalla de **WaveMaker** (Generador de olas). Hay varios submenús para esta opción particular. Esto es lo que aparece cuando se entra en esta sección por primera vez (la información que se presenta va a cambiar después de introducir tus preferencias):



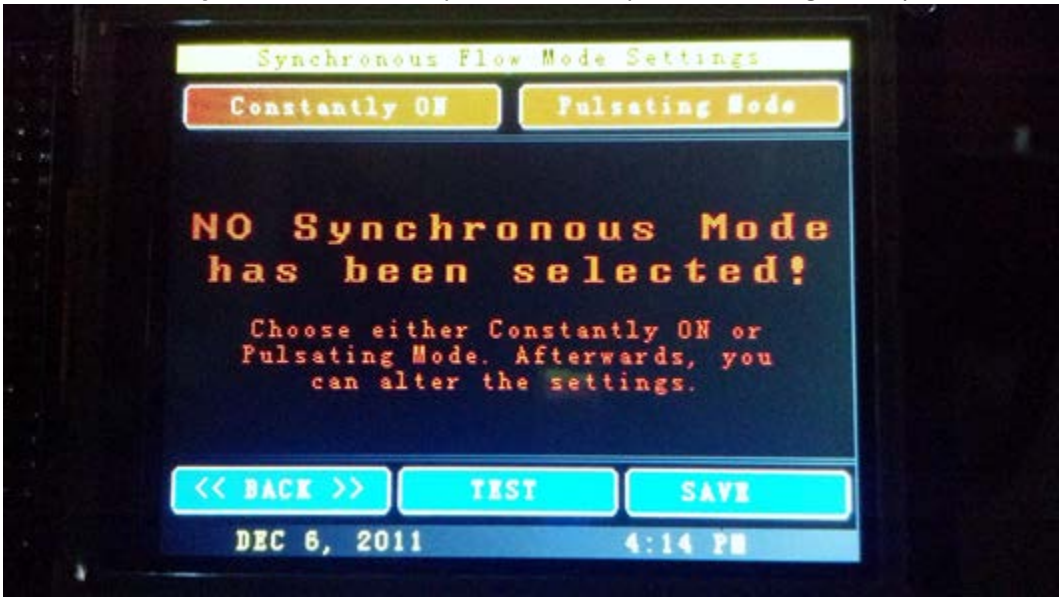
Seleccionando **Alterating Mode** (Modo alterno) de arriba, te lleva a la siguiente pantalla:



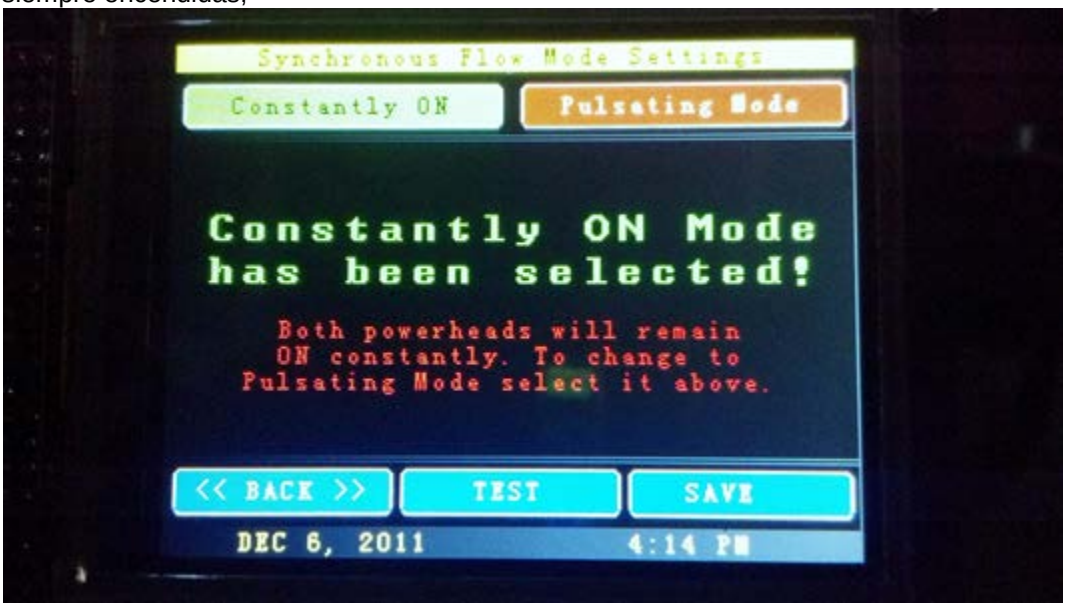
De aquí se puede establecer la duración de funcionamiento para cada bomba de movimiento, antes de cambiar a la otra. Puedes probar la configuración directamente desde esta pantalla. Al pulsar el botón TEST, este se volverá verde, y las bombas se encenderán y se apagaran acorde con el tiempo establecido. Si lo deseas, puedes guardarlo en la memoria EEPROM (memoria de Arduino). Después de guardar esto, al regresar a la pantalla principal de **WaveMaker**, verás lo siguiente:



Seleccionando **Synchronous Mode** (Modo Síncrono) te lleva a la siguiente pantalla:



Seleccionando **Constantly ON** (Constante) aparecerá lo siguiente: Ambas bombas de movimiento permanecerán siempre encendidas,



Y seleccionando la opción **Pulsating Mode** (Modo de pulsación) le llevará a esta página, lo que le permite programar el tiempo que las bombas de movimiento estarán en ON y OFF (encendida y apagada):

La configuración de esta página es casi lo mismo que configurar el modo alterno de arriba lo único que las dos bombas estarán encendidas y apagadas al mismo tiempo en vez de alternar de una a otra.

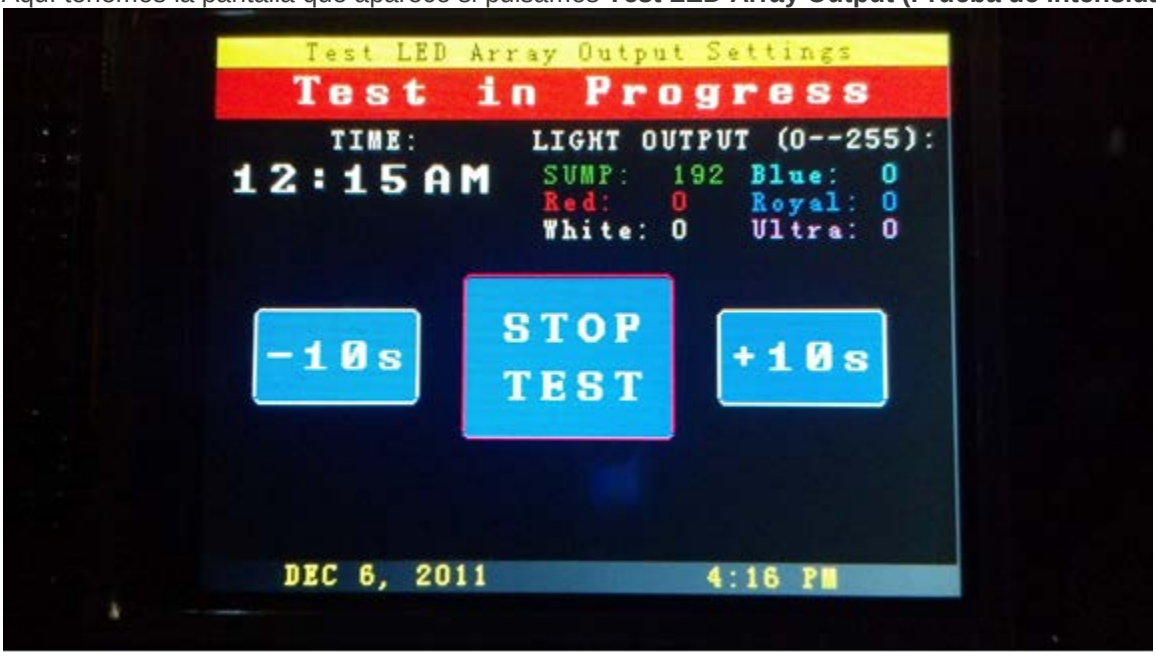
Y de vuelta a la pantalla principal de **WaveMaker** (Generador de olas), si se selecciona el **(Feeding Mode)** (Modo alimentación), verá este mensaje junto con una cuenta atrás: "*Las bombas permanecerán apagadas durante 5 minutos. El modo síncrono (modo que tenías seleccionado antes de pulsar Feeding Mode) se reanudara en: y la cuenta atrás."*

O presionando **Turn Pumps OFF** (Apagar bombas) aparecerá esto: *"Bombas Apagadas. Para volver a encender las bombas en modo síncrono, pulsar el botón* ***Turn Pumps ON*** *(Encender bombas) de arriba*

Presionando **CANCEL** o **BACK** también se reanudará el modo que teníamos seleccionado antes.

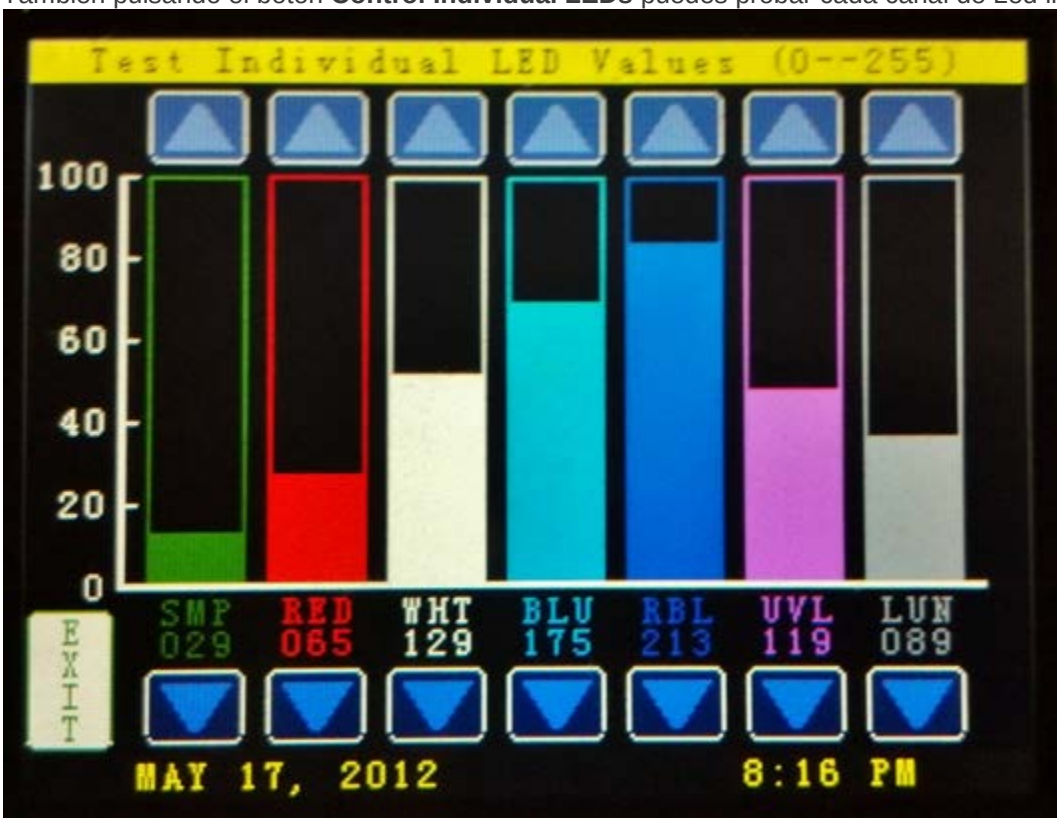
El siguiente paso en el MENÚ PRINCIPAL / pantalla **Choose Option** (Elegir Opciones) tenemos el botón **LED Testing (**Prueba de LED). Al pulsar esto le dará más opciones:

Aquí tenemos la pantalla que aparece si pulsamos **Test LED Array Output (Prueba de intensidad LED pantalla)**:



Esta prueba ejecuta el fotoperiodo de todo el día rápidamente, empezando de la medianoche. En otras palabras, esto te permite ver las intensidades de los LED´s y el cambio de iluminación en el avance rápido.

También pulsando el botón **Control Individual LEDs** puedes probar cada canal de Led individual.

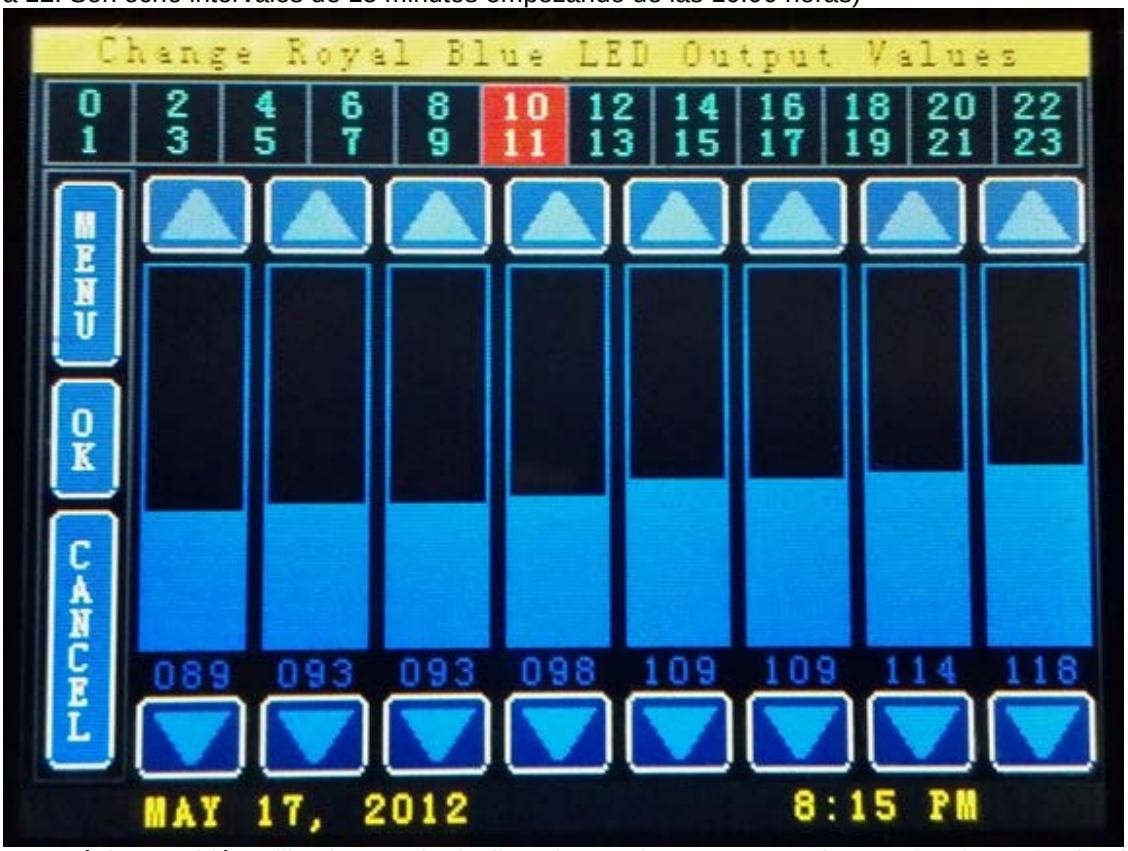


Lo que estamos viendo aquí son barras de deslizamiento táctiles de la pantalla y los botones. Puedes pulsar la barra, o arrastrar el dedo hacia arriba y hacia abajo para obtener rápidamente un valor, y los resultados son inmediatos. Incluso se puede deslizar el dedo en cualquier modelo y obtener ese resultado en todos los canales. Es necesario 1 segundo para que los colores vayan del 0 a los valores representados. Para un ajuste más preciso, puedes utilizar los botones arriba y abajo por encima de cada color. Una vez que pulsado "EXIT", saldrá de esta pantalla y los LED´s volverán a su horario regular.

Volviendo a la pantalla **CHOOSE OPTIONS** (Elegir Opciones), seleccionando **Change LED Values** (Cambiar valores de los LED´s) nos lleva a este menu:

Al seleccionar cualquiera de los colores o el botón de sumidero a continuación te llevará a esta página, que muestra los valores de LED en una franja de 24 horas. 0-1 son los valores de los LED´s entre medianoche 00 y 02 de la madrugada dividida en ocho cuartos de hora.

Al pulsar el botón **MORE COLORS** (Mas Colores) permitirá seleccionar entre los demás colores, al pulsar el botón **CHANGE** (Cambia) te permitirá cambiar el valor para cada período de tiempo (intervalos de 15 minutos). Seleccionado abajo los intervalos de 10 a 12 AM: (Recuerda que 10-11 significa el intervalo de 10 a 11 y de 11 a 12. Son ocho intervalos de 15 minutos empezando de las 10:00 horas)



Esta página también utiliza barras de deslizamiento y botones. Cuando se selecciona el color, las barras de deslizamiento y los números también será del mismo color que el canal de LED seleccionado.
Al pulsar **OK** le llevará de vuelta a la página con los nuevos valores. Al pulsar **SAVE** escribirá estos valores en la EEPROM, y se guardaran todos los valores. Después de un fallo eléctrico al reencender el Jarduino todas tus opciones

estarán guardadas.

Volviendo a la pantalla **Change LED Values** (Cambiar valores de los LED´s), si pulsamos el **Lunar,** nos llevara a esta pantalla:

Aquí puedes cambiar el valor máximo de iluminación del ciclo lunar (luna llena), así como el valor mínimo (también conocido como Luna Nueva). Tu acuario no tiene por qué ser totalmente oscuro cuando hay Luna Nueva si no quieres que sea.

Siguiente en la pantalla **Choose Options** (Elegir Opciones) verás la opción **Automatic Feeder** (Alimentador Automático). Este es otro menú que tiene varias opciones diferentes:

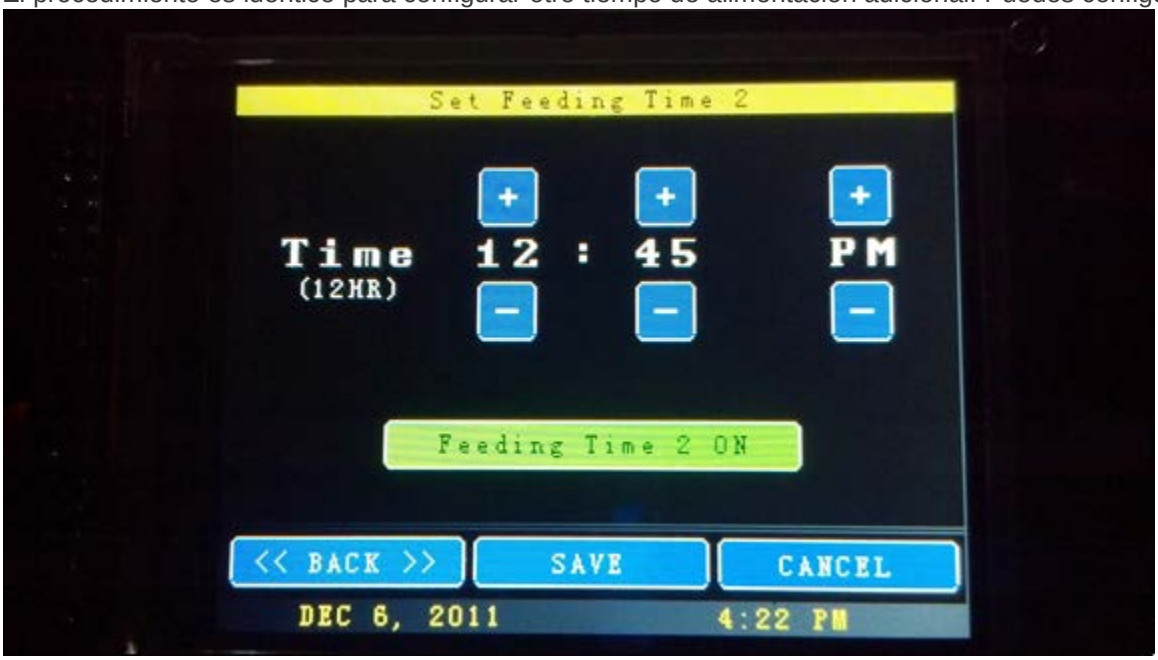
Esto es lo que aparece si no hay guardado ningún tiempo para alimentar a los peces.

Seleccionando **Feeding Time 1**, aparecerá esta pantalla:

Así que después de ajustar la hora, a continuación, decide si lo quieres activar o desactivar. Cuando se selecciona el botón rojo para activarlo, aparece esto:

Pulsando el botón **SAVE** nos llevara a la pantalla anterior:

El procedimiento es idéntico para configurar otro tiempo de alimentación adicional. Puedes configurar hasta cuatro.

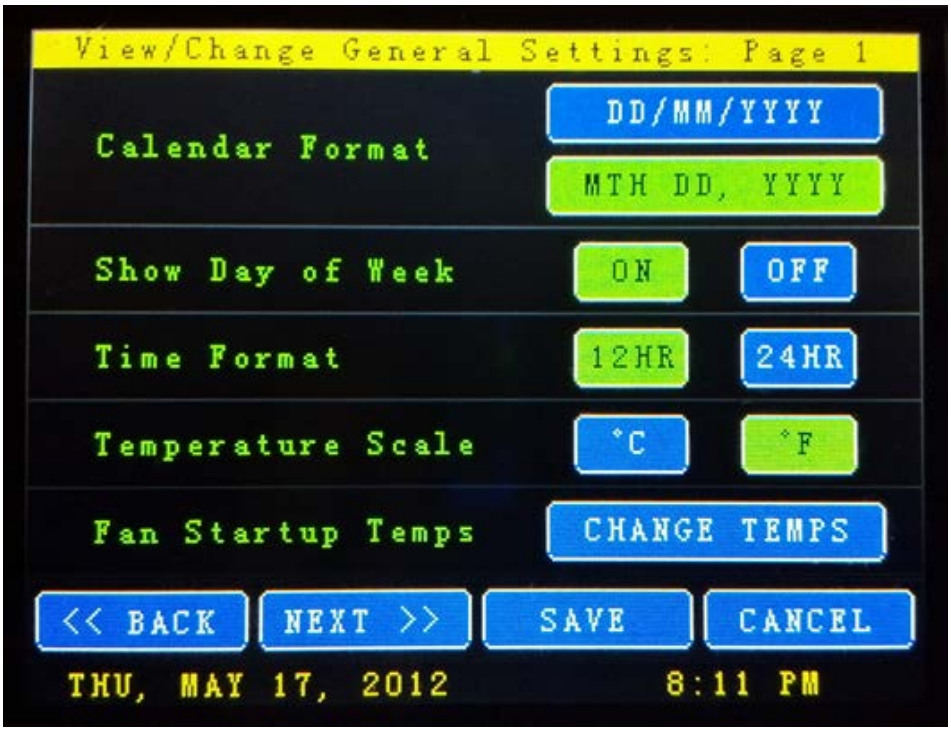


Otra foto de la pantalla con tres tiempos de alimentación configurados.

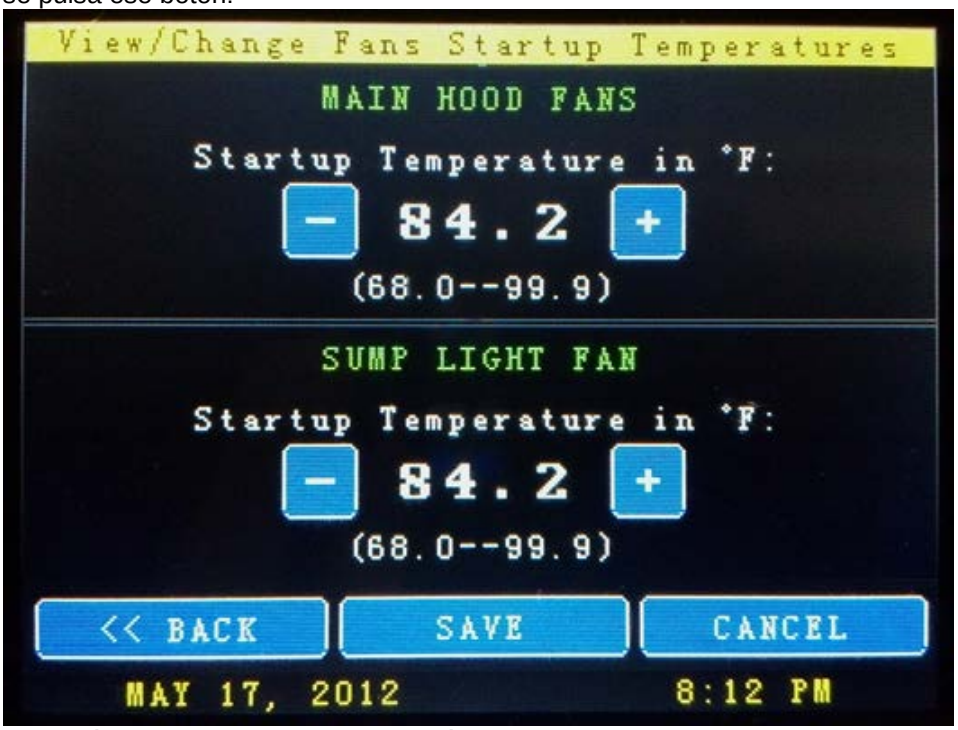
Creo que el botón violeta **Feed Fish Now!** (Alimentar a los peces ahora!) localizado en el centro de la pantalla no necesita explicación. Aquí está en acción:

Se mantiene verde durante unos segundos antes de volver al color violeta. Si quieres, puedes hacer que las bombas de movimiento se apaguen automáticamente durante 5 minutos, cuando hay tiempos de alimentación configurados o al pulsar el botón “Alimentar a los peces ahora!” Esto puede ser fácilmente programado seleccionándolo en la página **General Settings** (cubierto más adelante). Alternativo, puedes suspender las bombas de movimiento, desactivándolas en la página WaveMaker.

Por último, en **General Settings** (Configuración General) se presentan varias opciones fáciles de usar. Aquí está la página 1:

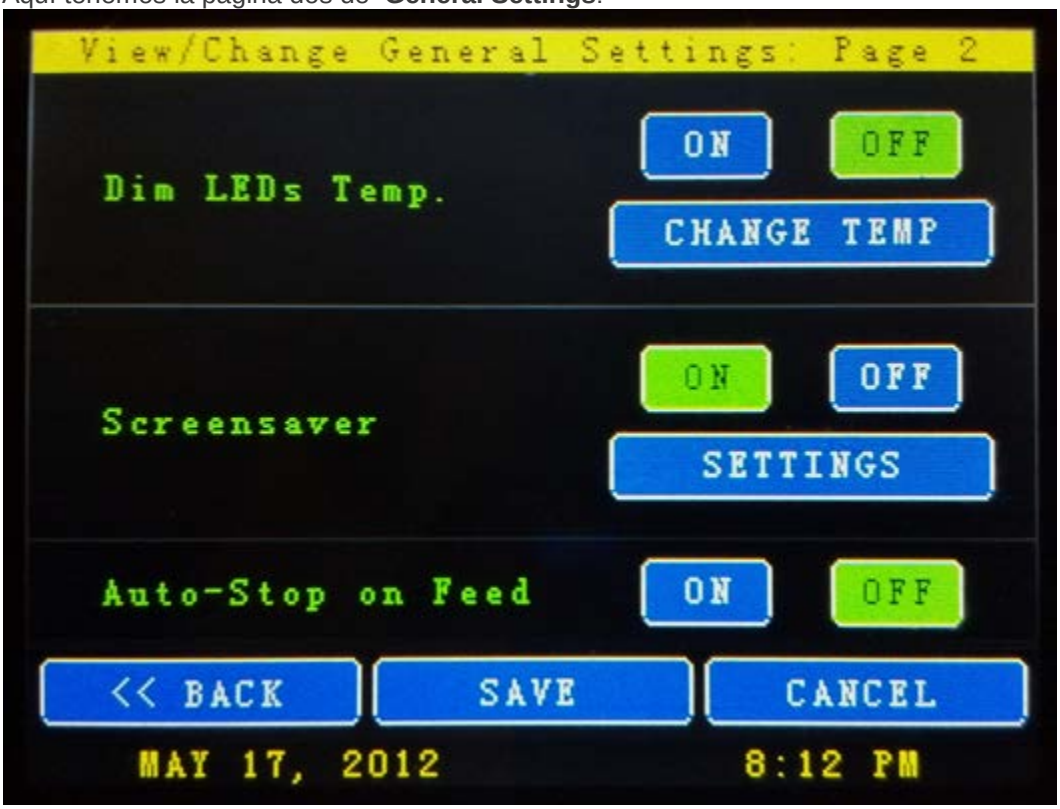


La mayoría de las cosas en este caso no necesitan explicación. Sólo tienes que seleccionar las opciones que te gustan, y cuando lo haces se resaltan en color verde. Pulse el botón "**SAVE**" y el Jarduino recordará tus selecciones. Una cosa que podría no ser tan obvio, es el último botón **Change Temps**. Aquí hay una captura de pantalla de lo que verás cuando se pulsa ese botón:



Esta página le permite elegir a qué temperatura los ventiladores PWM comenzarán a activarse / desactivarse. A partir de este valor, los ventiladores PWM aumentaran la velocidad un 10% por cada aumento de un grado de la temperatura.

Aquí tenemos la página dos de **General Settings**:



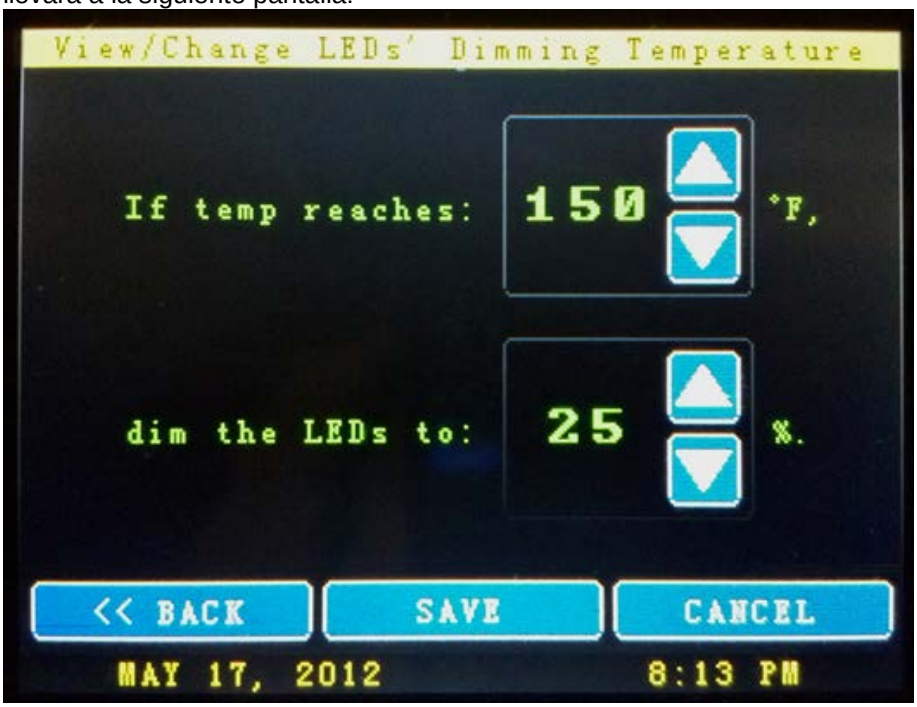
Hay tres opciones de esta página. Ahora puedes activar o desactivar de esta pantalla un protector de pantalla ("ON" o "OFF"). Seleccionando la opción **Settings** (Configuración) para Screensaver te traerá aquí:

Desde esta pantalla, puedes personalizar el protector de pantalla. En primer lugar, en la parte inferior ahora puedes elegir cuánto tiempo tardara para que el protector de pantalla se active. Puedes seleccionar entre uno y 99 minutos. Si decides que deseas es una pantalla en blanco **(Blank Screen)** como un protector de pantalla en lugar de un reloj digital, no hay problema. Sólo tienes que seleccionarlo. Pero si decides elegir el reloj, tiene la capacidad de mostrar

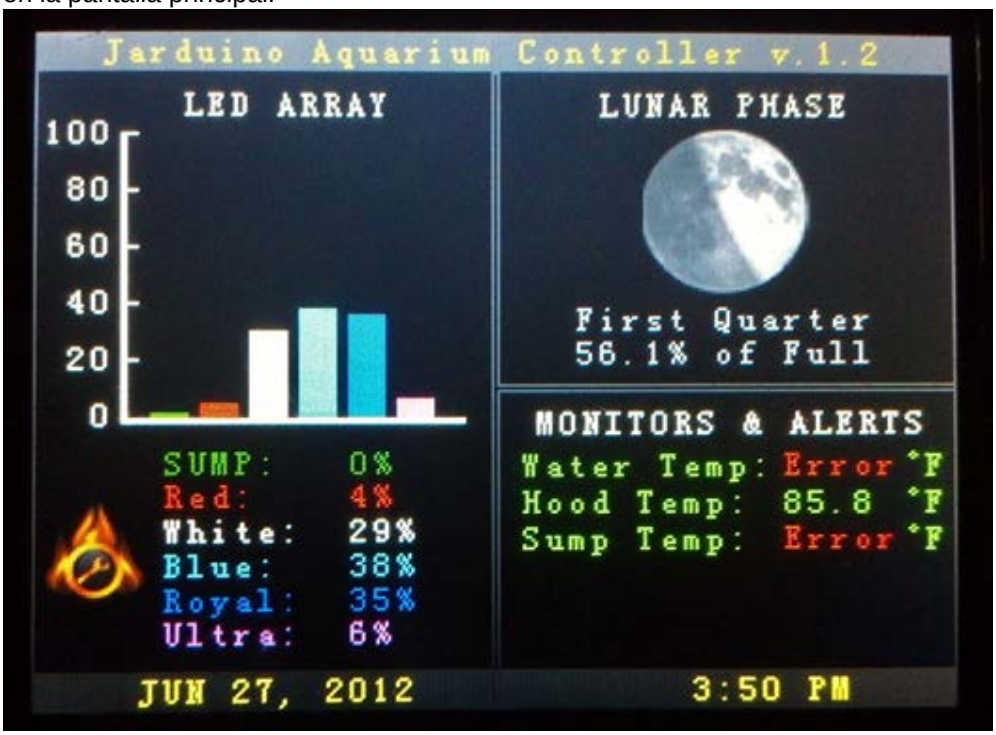
la fecha junto con la hora. Compruébalo tú mismo:

Por supuesto, cualquiera que sea el formato del calendario y de hora que elijas, también se trasladará a esta pantalla.

El **Dim LEDs Temp** (Dimeo de los LED con temperatura) es un sistema de protección incorporado que atenúa automáticamente los LED cada vez que el disipador alcanza una temperatura definida. Esta opción es para evitar que los LEDs se calienten demasiado, lo que dará lugar a un fallo prematuro. Al igual que el protector de pantalla, puedes elegir si deseas o no utilizar esta función seleccionando ON u OFF. Seleccionando **Change Temp** (Cambio de temperatura), te llevará a la siguiente pantalla:

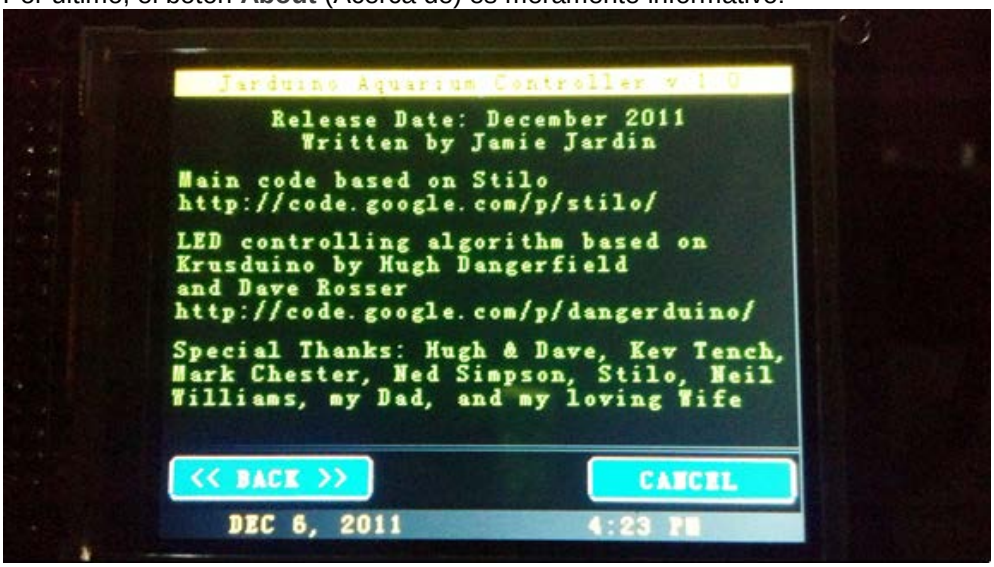


No hace falta decir por ahora, pero en caso de que te estas preguntando, la escala de temperatura que elegiste en la configuración general: Página 1, será trasladada a todas las funciones de temperatura. Lo que se hace aquí es seleccionar la temperatura a la que el protocolo de seguridad se active, para ilustrar los efectos, fue elegido el valor de 150 ° F (de un posible rango de 0-255). Sólo si la temperatura del disipador alcanza el valor seleccionado (La temperatura del disipador de calor es la que se controla, y la etiqueta es "Temp Hood" en la pantalla principal), entonces la intensidad de todos los LEDs se atenuará 25% de sus respectivos valores programados o cualquier valor que elijas, en cualquier valor de 0 a 100%, lo que por supuesto 0% no haría nada, y 100% lo atenuara a 0%, lo cual es completamente apagado). La intensidad de los LED´s seguirá siendo atenuada hasta que la temperatura baja 5 ° por debajo de la temperatura de control, momento en que se reanudará su horario normal. A no ser, por supuesto, llegar a esa temperatura definida, el ciclo se repite de nuevo. Si esta función se activa, aparecerá una pequeña llave en llamas en la pantalla principal:



La llama se convertirá en un color azul después de que la temperatura baja 5 º, momento en el que si lo tocas la pantalla, puedes descartar el "mensaje".

Por último, el botón **About** (Acerca de) es meramente informativo:



Gracias obligatorias a la esposa por darme la gran cantidad de paciencia, tiempo y comprensión para completar este proyecto!